



A THREE-DIMENSIONAL QUASI-NEUTRAL HYBRID
PARTICLE-IN-CELL CODE WITH APPLICATIONS TO THE
TILT MODE INSTABILITY IN FIELD REVERSED
CONFIGURATIONS: QN3D

Eric J. Horowitz
University of Maryland
Science Applications International Corporation

Dan E. Shumaker
David V. Anderson
National Magnetic Fusion Energy Computer Center
Lawrence Livermore National Laboratory

This paper was prepared for submittal to
JOURNAL OF COMPUTATIONAL PHYSICS

November 1987



Lawrence
Livermore
National
Laboratory

This is a preprint of a paper intended for publication in a journal or proceedings. Since changes may be made before publication, this preprint is made available with the understanding that it will not be cited or reproduced without the permission of the author.

DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

Classification Index Numbers: 77-A05

77-505

**Keywords: Gather, scatter, particle-in-cell, vectorization, interpolate,
Cray-2, sorting, multitasking, tilt mode, FRC, quasineutral,
hybrid, three dimensional**

Proposed running head: QN3D: A PIC Code and the Tilt Mode

Eric J. Horowitz

Laboratory for Plasma and Fusion Energy Studies

University of Maryland

College Park, MD 20742

Abstract

The tilt mode instability of a field reversed configuration (FRC) is discussed. Previous numerical models have not adequately explained the behavior of this mode. A particle-in-cell (PIC) model in Cartesian coordinates is introduced with an explanation as to why it represents the physics of FRC's more closely. The PIC model is implemented in a FORTRAN code, QN3D. The major elements of this code are presented including the many techniques required for its optimization. We discuss many of the major factors in optimization that are dependent upon features of the Cray-2 Multiprocessor. Testing of the code is presented in three phases. First, single particle motion is analyzed. Next the normal modes are calculated and simulated. Finally, QN3D is applied to the rigid rotor problem. These tests indicate that the code is suited to model plasma phenomena in the parameter regimes of interest. Lastly, two cases of the tilt mode problem are treated. The results match current experiments and confirm our initial hypothesis as to why other models are not adequate.

1 Introduction

This paper describes a 3-D hybrid, particle-in-cell code which is used to model the tilt mode in Field-Reversed Configurations (FRC's). FRC's are elongated compact toroidal plasmas which are formed in theta pinches with a static filling of neutral gas [1]. The FRC contains open and closed field line regions, with only poloidal fields.

Of particular interest is the $m = 1$ internal tilt which is an axial shift of the plasma with a toroidal mode number of 1. Figure 1 shows a plasma starting to tilt. Magnetohydrodynamics (MHD) codes have shown that the FRC should be violently unstable to the tilt mode with growth times comparable to the time it takes an Alfvén wave to transit the system. This is typically on the order of microseconds [2] and if the mode existed it would be very disruptive. Two-dimensional, non-linear MHD particle codes [3] [4] have been constructed to model this problem in a slab geometry. Simulations with these codes show that the tilt does not saturate but rather continues until the fields tear and reconnect resulting in the loss of particles and energy [5]. Other non-linear MHD codes confirm these results [6]. However, surprisingly enough, experiments which last for hundreds of microseconds (many MHD growth times) never exhibit this mode at all [1].

MHD codes categorically ignore the finite size of particle orbits about the magnetic fields, treating the entire plasma as a fluid that streams along the fields. A measure of the relative size of the ion orbits to the plasma scale length is s (see Eq. 12), which itself is a measure of the number of gyroradii between the o-point (field null) and the separatrix (separation between open and closed field lines). In low s cases (like present experiments), the ion orbits are large and could sample good magnetic field curvature possibly resulting in stability of the tilt mode. For high- s cases (like future experiments), the ions are more confined to the magnetic field lines which could result in instability. If this is the case, then the pessimistic predictions of MHD codes are applicable to larger experiments.

Therefore, we need a computational model that includes finite ion orbits. A code built on this model would have to run in both regimes to demonstrate consistency with current experiments and to predict the evolution of plasmas in larger experiments.

One of the first attempts to include ion motion was a modification of the MHD equations to include the ion orbital (Larmor) radius about the field lines [7]. This finite Larmor radius (FLR) model has two basic flaws – first, it does not account for very large radii of particles near the o-point, and second it assumes that the magnetic moment of the particles is an adiabatic invariant which was shown not to be the case [8].

A new method developed by Barnes, et al. [9] is based on the Vlasov-fluid model [10][11][12] which treats ions as collisionless particles and the electrons as a cold and massless fluid. A dispersion functional is developed [13] which involves time history integrals for the ion trajectories. These integrals are evaluated numerically by using many ion orbits determined by a self-consistent equilibrium. Using the MHD eigenfunction as the trial function in the dispersion functional, eigenfrequencies can be computed which include the kinetic ion effects. This should lead to a computation of the linear growth rates for all of the modes. This work is still in progress.

1.1 Introduction to QN3D

We present a code that explicitly follows many ion orbits and computes fields consistent with ion motion. Also, we make no linearizing assumptions. But, since the gyroradii of electrons is about one fortieth that of ions, we still ignore their effect and treat the electrons as an inertialess fluid. The purpose of this work was to develop such a code and to apply it to the tilt mode problem. In addition to correctly modelling the tilt mode in the linear regime, our code should be able to follow the evolution of the mode into non-linear regimes.

Since the tilt mode is a three dimensional phenomenon, the code developed, QN3D is also three dimensional. Such a code was not feasible before the advent of the Cray-2 computer since large amounts of storage are required. Also, certain essential phases of computation had been very slow prior to the introduction of some special hardware instructions on the Cray-2.

In the remainder of this paper we first discuss the parameter regime of interest from which we derive a basic set of equations. Next, in section 2 we explain the computational cycle. In section 3 we explore single particle motion in static fields.

This is followed by normal mode simulations in section 4 and the rigid rotor instability in section 5. Finally we explore the tilt mode in section 6.

To date, we have simulated a few cases that illustrate QN3D's suitability to the problem and test our initial hypothesis regarding larger machines and smaller orbits.

Starting with the standard Maxwell's equations and the equations of motion for particles, we have derived a final set of equations. Maxwell's equations, in the Gaussian system of units, are

$$\begin{aligned}\nabla \times \mathbf{B} &= \frac{4\pi}{c} \mathbf{J} + \frac{1}{c} \frac{\partial \mathbf{E}}{\partial t} \\ \nabla \times \mathbf{E} &= -\frac{1}{c} \frac{\partial \mathbf{B}}{\partial t} \\ \nabla \cdot \mathbf{E} &= 4\pi \rho_c \\ \nabla \cdot \mathbf{B} &= 0.\end{aligned}$$

The ion motions will be solved explicitly using the equation of motion which is

$$M \frac{d\mathbf{v}_i}{dt} = e\mathbf{E} + \frac{e}{c} \mathbf{v}_i \times \mathbf{B}$$

A fluid equation will be used for the electrons which we assume to be inertialess and cold.

$$0 = -e(\mathbf{E} + \frac{\mathbf{u}_e \times \mathbf{B}}{c})$$

An additional assumption of quasineutrality ($n_i \approx n_e$) implies

$$\nabla \cdot \mathbf{E} = 4\pi e(n_i - n_e) \approx 0 \quad \Rightarrow \quad \mathbf{E}_l \approx 0.$$

The subscript, l , refers to the longitudinal component. Conservation of charge with $n_i \approx n_e$ yields $\mathbf{J}_l \approx 0$.

Finally, light waves are omitted by assuming that they propagate instantaneously. This is called the Darwin model [14] and it is common for particle-in-cell codes that follow ion motion. The effect is to ignore the transverse displacement current in Ampère's Law, i.e.

$$\nabla \times \mathbf{B} = \frac{4\pi}{c} \mathbf{J} + \frac{1}{c} \frac{\partial \mathbf{E}_l}{\partial t},$$

which is equivalent to

$$\nabla \times \mathbf{B} = \frac{4\pi}{c} \mathbf{J}_l \quad \text{and} \quad 0 = \frac{4\pi}{c} \mathbf{J}_l + \frac{1}{c} \frac{\partial \mathbf{E}_l}{\partial t}.$$

The last equation is solved identically if we let $\mathbf{E}_l = \mathbf{J}_l = 0$.

The assumptions mentioned above can be shown to be consistent with the physical process we are modelling by expanding the equations in a small parameter $\epsilon (= v_A/c)$ [15].

The results of our assumptions are the following equations for the electromagnetic fields, the electron current and the ion motion –

$$\nabla \times \mathbf{E} = -\frac{1}{c} \frac{\partial \mathbf{B}}{\partial t} \tag{1}$$

$$\nabla \times \mathbf{B} = \frac{4\pi}{c} \mathbf{J} \tag{2}$$

$$\mathbf{E} = \frac{\mathbf{J}_e \times \mathbf{B}}{en_i c} \tag{3}$$

$$M \frac{d\mathbf{v}_i}{dt} = e\mathbf{E} + \frac{e}{c} \mathbf{v}_i \times \mathbf{B}, \tag{4}$$

where $\mathbf{J}_e = -en_i \mathbf{u}_e$. Notice that we have set $n_e = n_i$ consistent with quasineutrality. Also note that this system is non-linear due to Eqs. 3 and 4. QN3D integrates these equations in time and the method of solution is described in section 2.

Notice that Eq. 3 becomes ill-defined as $n_i \rightarrow 0$. But even before this happens our scheme becomes incorrect. The final equations were derived under the assumption that ϵ is small. But as $n_i \rightarrow 0$, $v_A \rightarrow \infty$ and ϵ becomes very large, negating our premise. Thus we solve $\nabla^2 \mathbf{E} = 0$ in the low-density region.

Two of the original four Maxwell's equations are not represented here. The first is Gauss' Law, $\nabla \cdot \mathbf{E} = 4\pi\rho_e$. The only contribution of the density is in Eq. 3 where we use $n_e \approx n_i$ to zeroth order. Gauss' law could be used to find n_e^1 but it does not contribute to the system of equations, and thus we never solve it.

The second is $\nabla \cdot \mathbf{B} = 0$. But if $\nabla \cdot \mathbf{B}$ is initially zero, then it always will be, and we can eliminate it from the final set of equations.

2 The Computational Cycle

We use a particle-in-cell (PIC) model for two reasons. The first is that since a plasma is an ionized gas, the particles shield each other within a Debye length [16]. Thus particle-particle interactions are not effective over spacial scales longer than λ_D . The second is that the collision frequency (ν_c) is very much smaller than the frequency of long-range plasma effects such as the plasma frequency (ω_p) [16]. Thus we can ignore particle-particle interactions when we are considering phenomena occurring on space scales longer than λ_D and on time scales shorter than ν_c^{-1} .

The PIC method consists of particles moving in free space under the influence of fields that are computed on a grid. Thus, given electromagnetic fields computed on a grid, they are interpolated to the particles, the particles are moved according to the forces exerted on them by these fields and then the charge and current densities are interpolated to the grid. To end the cycle the fields are solved on the grid consistent with the charge and current densities to prepare for the next cycle (see Fig. 2). The rest of this chapter addresses each phase of the PIC cycle individually, stressing techniques used to take advantage of the Cray-2.

2.1 Solving the Field Equations

The particle push gives us \mathbf{J}_i and n_i at an advanced time step through a time-explicit integration of the ion equations of motion (see section 2.3). If we are currently at time step n , then these particle quantities will be at step $n + 1/2$ or $n + 1$ depending upon the needs of the fields solving algorithm. The function of the field solver is to advance the field quantities \mathbf{E} and \mathbf{B} from step n to step $n + 1$ consistent with the ion quantities.

A straightforward method for solving the fields is functional iteration. Upon a suggestion from Hewett [17], we developed a time and space centered iteration scheme to advance the fields from step n to $n + 1$ using particle quantities at time step $n + 1/2$.

Discretizing in time, Eqs. 1-3 become

$$\mathbf{B}_{k+1}^{n+1} = \mathbf{B}^n - \frac{c\Delta t}{2} \nabla \times (\mathbf{E}_k^{n+1} + \mathbf{E}^n)$$

$$\begin{aligned}
\mathbf{J}_{e,k+1}^{n+1/2} &= \frac{c}{4\pi} \frac{1}{2} \nabla \times (\mathbf{B}_{k+1}^{n+1} + \mathbf{B}^n) - \mathbf{J}_i^{n+1/2} \\
\mathbf{E}_{k+1}^{n+1} &= \frac{1}{en_i^{n+1/2} c} \mathbf{J}_{e,k+1}^{n+1/2} \times (\mathbf{B}_{k+1}^{n+1} + \mathbf{B}^n) - \mathbf{E}^n.
\end{aligned} \tag{5}$$

Since the first of these equations requires a knowledge of \mathbf{E} at the new time step, we guess at its value and find a better guess from the last equation. Thus this is an iterative scheme with the subscript k indicating the iteration.

All spacial derivatives are represented as central differences. To initialize the iteration we used $\mathbf{E}_i^{n+1} = \mathbf{E}^n$. It characteristically converged in 4-9 iterations to a relative error of 10^{-3} .

The scheme presented above has a limitation in that it is sensitive to the particle density. There it was mentioned that the assumptions about the expansion parameter, ϵ were violated. In addition, long before $n_i = 0$, any explicit numerical scheme will become unstable since the CFL condition is violated. In this case the CFL condition is approximately $\Delta t \leq \Delta_x/v_A$, a standard result for hyperbolic systems. In certain cases, the ion response can be ignored and a parabolic system results. Then the CFL condition is $\Delta t < \frac{1}{2}\Omega\Delta_x^2/v_A^2$ which is typically more stringent than the hyperbolic condition. To avoid this problem, $\nabla^2\mathbf{E} = 0$ is solved in the low-density regions.

To solve the vacuum equation, a standard Gauß-Seidel method could be employed. For example, the x compoinent of \mathbf{E} is

$$\begin{aligned}
EX(I,J,K) = & DELF*((EX(I+1,J,K)+EX(I-1,J,K))*DXIS+ \\
& (EX(I,J+1,K)+EX(I,J-1,K))*DYIS+ \\
& (EX(I,J,K+1)+EX(I,J,K-1))*DZIS),
\end{aligned}$$

where $DELF = 1/(2*(DXIS+DYIS+DZIS))$ and $DXIS = 1/\Delta_x^2$, etc. The notation has changed to indicate that this is a FORTRAN assignment statement which gives us an iteration algorithm for solving for EX.

The value of the electric field is assumed to be given on the external boundary and in the plasma. For each iteration of Eqs. 5 the electric field is found in the low-density regions by this method.

2.1.1 Multigrid Method

Unfortunately, Gauß-Seidel converges very slowly and improvements such as successive overrelaxation offer only minor improvement. To speed up convergence, a multigrid method was employed.

The multigrid method has many advantages. For instance, it does not require symmetric operators as do conjugate gradient methods, nor does it require a lot of additional storage as do most direct methods. But mostly, it is very much faster than other iterative methods.

The basic idea of the method is to use iterative schemes in the limited regions where they work best. Methods like Gauß-Seidel reduce short wavelength errors very effectively but do very poorly on long wavelength errors. But, on a coarser grid, relatively long wavelength errors become relatively short wavelength errors (relative to the current grid spacing) and then iterative methods become more effective.

For computational simplicity we choose coarser grids which are subsets of the original, finest grid. For each coarser grid, we use every other point in each of the three spacial directions of the finer grid. Thus each coarser grid has one eighth the grid points of the previous, finer grid. As a result, we can reduce the work by a factor of 512 with only four levels of grids. One can begin to see why multigrid is so promising for a three-dimensional problem.

The most obvious scheme would be to solve the given equation on every grid. In fact, we do solve $\nabla^2 \mathbf{E} = 0$ on the finest grid, but on the other grids we solve for the error of the previous, finer grid. This method, called Cycle C, is due to Achi Brandt [18]. The idea is to solve

$$\mathbf{L}\mathbf{U} = \mathbf{S} \quad (\nabla^2 \mathbf{E} = 0 \text{ in our case}),$$

where \mathbf{L} is some linear operator, \mathbf{S} is some given source and \mathbf{U} is the quantity we want to find. We will denote the discretized form of this equation as

$$\mathbf{L}_1 \mathbf{U}_1 = \mathbf{S}_1, \tag{6}$$

where the subscript 1 refers to the grid number with higher numbers referring to

coarser grids in our notation. We can estimate U_1 by applying the iteration once

$$\widetilde{U}_1 = \text{relax}(L_1 U_1 = S_1),$$

where “relax” is multigrid lingo for applying the iteration once. The error of \widetilde{U}_1 , given by \overline{U}_1 , is defined by

$$\begin{aligned} U_1 &= \widetilde{U}_1 + \overline{U}_1 \Rightarrow \\ L_1 U_1 &= L_1 \widetilde{U}_1 + L_1 \overline{U}_1 = S_1 \Rightarrow \\ L_1 \overline{U}_1 &= S_1 - L_1 \widetilde{U}_1 \doteq R_1. \end{aligned}$$

where the ‘ \doteq ’ notation denotes ‘defined as’. If we can find \overline{U}_1 then we have U_1 . But instead of solving $L_1 \overline{U}_1 = R_1$, we solve an analogous equation on a coarser grid:

$$L_2 U_2 = S_2,$$

where $S_2 = I_1^2 R_1$ (I_j^k is the interpolation operator from grid j to k -- see below). We choose to call the source of this equation S_2 rather than R_2 and the field U_2 rather than \overline{U}_2 so that this equation will resemble equation 6. Now we solve this equation just like the previous one, i.e.

$$\begin{aligned} \widetilde{U}_2 &= \text{relax}(L_2 U_2 = S_2) \\ U_2 &= \widetilde{U}_2 + \overline{U}_2 \Rightarrow \\ L_2 \overline{U}_2 &= R_2 \quad (R_2 = S_2 - L_2 \widetilde{U}_2). \end{aligned} \tag{7}$$

Note that \overline{U}_2 is the error of U_2 which itself is the coarse grid analog of \overline{U}_1 , the error of U_1 . In general, on each grid we solve for the error of the previous grid. Thus, again we choose to solve equation 7 on a coarser grid. This process continues to the coarsest grid, M , where we have

$$\begin{aligned} L_M U_M &= S_M \quad (S_M = I_{(M-1)}^M R_{(M-1)}) \\ \widetilde{U}_M &= \text{relax}(L_M U_M = S_M). \end{aligned}$$

But now we relax many times until

$$\begin{aligned}\widetilde{\mathbf{U}}_M &\approx \mathbf{U}_M \Rightarrow \\ \overline{\mathbf{U}}_M &\approx 0 \quad \text{and} \\ \mathbf{R}_M &\approx 0.\end{aligned}$$

To go back up to finer grids we set

$$\begin{aligned}\overline{\mathbf{U}}_{(M-1)} &= \mathbf{I}_M^{(M-1)}(\mathbf{U}_M) \Rightarrow \\ \mathbf{U}_{(M-1)} &= \widetilde{\mathbf{U}}_{(M-1)} + \mathbf{I}_M^{(M-1)}(\mathbf{U}_M) \Rightarrow \dots \\ \mathbf{U}_1 &= \widetilde{\mathbf{U}}_1 + \mathbf{I}_2^1(\mathbf{U}_2).\end{aligned}$$

Since this \mathbf{U}_1 might only be approximately right, we use it to relax ($\mathbf{L}_1 \mathbf{U}_1 = \mathbf{S}_1$) and start the cycle again. This process is outlined in Fig. 3, which includes the slight variations of relaxing on each grid on the way up (coarser to finer) as well as on the way down, and relaxing any number of times on a given grid. Going from the finest grid to the coarsest grid and back up again as described here is called a V-cycle.

Interpolating or projecting quantities from the fine grid to the coarse grid is done by taking a weighted average of the values at the coarse grid point and the points of the finer grid surrounding it. Thus $\mathbf{U}_2 = \mathbf{I}_1^2(\mathbf{U}_1)$ would be

$$\begin{aligned}\text{UCG}(I,J,K) &= (1/12) * (6 * \text{UFG}(I,J,K) + \\ &\quad \text{UFG}(I+1,J,K) + \text{UFG}(I-1,J,K) + \\ &\quad \text{UFG}(I,J+1,K) + \text{UFG}(I,J-1,K) + \\ &\quad \text{UFG}(I,J,K+1) + \text{UFG}(I,J,K-1)),\end{aligned}$$

where UFG is \mathbf{U}_1 and UCG is \mathbf{U}_2 . Such an arrangement is called a star [19] since a template placed on the grid to this effect would look like a star.

To interpolate from coarser to finer grids we use tri-linear weighting.

To complete the algorithm we need to know when to stop. The multigrid method is very convenient since it is simple to estimate when the method will no longer be effective. The basic idea is that there is a local truncation error (LTE) incurred in interpolating between grids. When the residual ($\mathbf{R}_1 \equiv \mathbf{L}_1 \mathbf{U} - \mathbf{S}_1$) becomes less than the LTE then nothing more can be gained by further iterations. A simple estimate of the LTE is given by [20]

$$\text{LTE} = \mathbf{L}_2(\mathbf{I}_1^2 \mathbf{U}_1) - \mathbf{I}_1^2(\mathbf{L}_1 \mathbf{U}_1).$$

This is simply the difference between operating on the interpolated field and interpolating the operated field. In particular, we compare the L_2 norms of the LTE and \mathbf{R}_1 , i.e.

$$\|\mathbf{R}_1\|_2 \doteq \left[\int \mathbf{R}_1^2 d\mathbf{x} \right]^{1/2}$$

which is discretized appropriately. With this type of convergence check, we do not need to specify a minimum value of $\|\mathbf{R}_1\|_2$.

2.1.2 Plasma - Vacuum Interface

There is a great advantage to solving the error on each grid rather than the field itself for our particular application. We have an internal plasma region where the field is given by the solving equations 5. This is an internal boundary which might not be resolved well by coarser grids. Thus the value of the fields at the internal boundaries could change drastically as we go to coarser grids (see Fig. 4). However, if we are solving for the error in the fields, then we assume that the field is correct within the plasma and thus the error, $\bar{\mathbf{U}}_m$, is always zero there.

2.1.3 Results

We ran several rigid-rotor problems (see chapter 5) to compare with the standard SOR scheme. Both schemes gave the same growth rates of the rotational instability. But also, the computer time required by the multigrid method was only slightly less than that required for standard SOR. However, when we tried a larger problem with 41 grid points in each direction and 800,000 particles, multigrid was 6.4 times faster than SOR.

2.2 Interpolating the Electromagnetic Fields from the Grid to the Particles

Once the fields exist on the grid, we must interpolate them to the particle positions. QN3D uses tri-linear weighting to do this. We will discuss bi-linear weighting here since two dimensional pictures are easier to draw and understand than three

dimensional ones, but the three-dimensional analog to this discussion should be quite evident.

The two-dimensional formula for bi-linear interpolation is

$$\begin{aligned} \text{BPX}(N) = & A1 * \text{BGX}(\text{IP}(N), \text{JP}(N)) + A2 * \text{BGX}(\text{IP}(N)+1, \text{JP}(N)) + \\ & A3 * \text{BGX}(\text{IP}(N), \text{JP}(N)+1) + A4 * \text{BGX}(\text{IP}(N)+1, \text{JP}(N)+1). \end{aligned}$$

where $\text{BPX}(N)$ is the x component of the magnetic field at the position of particle N , $\text{BGX}(I,J)$ is the x component of the magnetic field at grid point I,J and $A1$ through $A4$ are the areas shown in Fig. 5.

Similar equations hold for the other five components of the electromagnetic fields. In three dimensions, the above equation would have eight terms corresponding to the volumes that represent the overlap of the particles with the domains of its eight nearest grid points.

We notice that the indices of BGX , IP and JP have their own index, N . This construct is called indirect indexing. Computers such as the Cray-2 have hardware instructions to allow the vectorization of these constructs with a sizeable reduction in the CPU time required for similar tasks [21]. The concepts, *gather* and *scatter*, respectively refer to the techniques used to read and write BGX from and to memory.

2.3 Advancing the Particles

Once the fields are known at the particle positions, the particles are advanced from their old positions to their new positions by integrating the equations of motion. The equations of motion are (suppressing the ion subscript, i)

$$\frac{d\mathbf{v}}{dt} = \frac{\mathbf{f}}{M} \quad \text{and} \quad \frac{d\mathbf{r}}{dt} = \mathbf{v},$$

where \mathbf{v} is the velocity, M is the mass and \mathbf{r} is the position of a particle. \mathbf{f} is the total force on the particle and t is time.

QN3D integrates these equations using a leap-frog scheme:

$$\frac{\mathbf{v}^{n+1/2} - \mathbf{v}^{n-1/2}}{\Delta t} = \frac{\mathbf{f}^n}{M}, \tag{8}$$

$$\mathbf{r}^{n+1} = \mathbf{r}^n + \Delta t \mathbf{v}^{n+1/2}.$$

where \mathbf{f}^n is the Lorentz force

$$\mathbf{f}^n = e(\mathbf{E}^n + \frac{\mathbf{v}^n \times \mathbf{B}^n}{c}).$$

Using this form of \mathbf{f}^n appears to make Eq. 8 an explicit expression for $\mathbf{v}^{n+1/2}$ but since \mathbf{v} is known only at half time steps, \mathbf{v}^n is defined as

$$\mathbf{v}^n = \frac{(\mathbf{v}^{n+1/2} + \mathbf{v}^{n-1/2})}{2}.$$

Thus,

$$\mathbf{v}^{n+1/2} = \mathbf{v}^{n-1/2} + \frac{e\Delta t}{M}(\mathbf{E}^n + \frac{(\mathbf{v}^{n+1/2} + \mathbf{v}^{n-1/2}) \times \mathbf{B}^n}{2c}), \quad (9)$$

which includes implicit terms on the right-hand side. Moving the implicit terms to the left gives a matrix equation of the form

$$\mathbf{A}^n \cdot \mathbf{v}^{n+1/2} = \mathbf{S}^n.$$

Fortunately, \mathbf{A}^n *always* has an inverse given by

$$(\mathbf{A}^n)^{-1} = \frac{1}{\delta} \begin{pmatrix} 1 + \alpha^2 & \alpha\beta + \gamma & \alpha\gamma - \beta \\ \alpha\beta - \gamma & 1 + \beta^2 & \beta\gamma + \alpha \\ \alpha\gamma + \beta & \beta\gamma - \alpha & 1 + \gamma^2 \end{pmatrix}$$

where

$$\alpha = \frac{e\Delta t}{2Mc} B_x^n,$$

$$\beta = \frac{e\Delta t}{2Mc} B_y^n,$$

$$\gamma = \frac{e\Delta t}{2Mc} B_z^n$$

$$\text{and } \delta = 1 + \alpha^2 + \beta^2 + \gamma^2.$$

Thus the equation that solves $\mathbf{v}^{n+1/2}$ is

$$\mathbf{v}^{n+1/2} = (\mathbf{A}^n)^{-1} \cdot \mathbf{S}^n,$$

which is now completely explicit. Now the positions can be advanced by solving

$$\mathbf{r}^{n+1} = \mathbf{r}^n + \Delta t \mathbf{v}^{n+1/2}.$$

After this computation, the velocities are known at time step $n + 1/2$ and the positions are known at time step $n + 1$. However, the field solve requires the charge and current densities at the $1/2$ integer time step, thus the particles must be pushed only one-half time step from their previous positions, i.e.

$$\mathbf{r}^{n+1/2} = \frac{1}{2}(\mathbf{r}^{n+1} + \mathbf{r}^n),$$

which is equivalent to

$$\mathbf{r}^{n+1/2} = \mathbf{r}^n + \frac{\Delta t}{2} \mathbf{v}^{n+1/2}.$$

After the field solve, the particles are advanced to the full time step to preserve the accuracy of the scheme.

Since the leap-frog scheme is explicit, it can be vectorized directly by the compiler.

Note that we do not employ the “ $\tan \alpha / \alpha$ ” correction mentioned by Birdsall and Langdon [22]. This is because our orbits are so complicated due to the nature of the magnetic fields, that we will use a time step that is small relative to the gyroperiod to resolve them. In this regime, our scheme incurs only a small phase error in the particle trajectories due to the neglect of this correction.

It is straight forward to show that this scheme conserves energy identically in the case of no electric fields and time invariant magnetic fields. Using Eq. 9 for $\mathbf{v}^{n+1/2}$, considerable algebraic manipulation gives explicitly

$$\mathbf{v}^{n+1/2} \cdot \mathbf{v}^{n+1/2} = \mathbf{v}^{n-1/2} \cdot \mathbf{v}^{n-1/2}.$$

Thus the numerical algorithm conserves energy identically for the static case.

Two time-step restrictions originate from this method. The first is that scheme must resolve the ion cyclotron motion about the magnetic field lines. The cyclotron period is

$$t_c = 2\pi \frac{Mc}{eB}.$$

Thus $\Delta t \ll t_c$ to resolve ion orbits. We normally used $\Delta t < .1t_c$.

The second restriction is that a particle should not travel more than one grid cell per time step. Otherwise, the assumption that the fields are relatively constant over the trajectory length traversed in one time step is violated. Thus $\Delta t < v_x \Delta x$ with similar restrictions in the other directions. These restrictions are characteristically not as stringent as the CFL conditions.

2.4 Interpolating Charge and Current Densities from the Particles to the Grid

To find the charge and current densities on the grid, we reverse the procedure described in section 2.2 and obtain similar coding, except for some surprises. A first attempt at the deposition might look like:

```
DO 20 N=1,NMAX
    A1 = (XG(IP(N)+1)-XP(N))*(YG(JP(N)+1)-YP(N))/A
    A2 = ...
    A3 = ...
    A4 = ...
    D(IP(N),JP(N)) = D(IP(N),JP(N))+A1*Q/A
    D(IP(N)+1,JP(N)) = D(IP(N)+1,JP(N))+A2*Q/A
    .
    .
    .
20 CONTINUE
```

where $D(I,J)$ is the charge density at grid point (I,J) and Q is the particle charge. $NMAX$ is the total number of particles and A is the area of a grid cell. This loop represents the contribution of a particle to the charge density of the four grid points surrounding it. Note that D appears on both sides of the equation since we are computing the running sum of contributions over all particles that are in any given cell.

The problem with vectorizing such a loop is that several particles can contribute to the density of a particular grid point, and thus an element of the array $D(I,J)$ will be assigned several values as we sweep through the particles. The compiler will note

this point and will not vectorize the above loop.

To circumvent this problem, we can sort the particles into groups such that within a group, no two particles occupy the same cell (see Fig. 6). The sorting algorithm requires a one-pass sweep through the particles in a non-vectorized loop. It is presented in detail in reference [21] where we cited a factor of three speed up when compared to scalar deposition.

2.5 Multitasking

QN3D has been designed to take advantage of the multitasking capability of the Cray-2. We have been able to multitask each phase of the computational cycle and discover some of the key issues involved in doing so in a time-sharing environment. Multitasking is judged by the CPU overlap, \mathcal{M} , which is defined as [23]

$$\mathcal{M} = \frac{\tau_c}{\tau_p}$$

where τ_c is the total CPU time used and τ_p , the processing time, is the time during which *at least* one CPU was occupied. Since a code must compete for resources with other codes, the average overlap is a statistical quantity dependent on many factors. QN3D has been helpful in understanding some of the issues involved in obtaining satisfactory overlap in a time-sharing environment.

Parallelizing most segments are straightforward. If we use an explicit field solve then each grid point gets updated field information which is independent of every other grid point, and thus the order in which they are done is irrelevant and they can be done simultaneously. Similarly, when we interpolate the field quantities from the grid to the particles, the fields at a particular particle's position is independent of fields at any other particle, so again the job of assigning fields can be delegated to several tasks which can run simultaneously. The same argument applies to advancing the particles.

The multigrid algorithm which solves $\nabla^2 \mathbf{E} = 0$ in the low density regions (section 2.1.1) is also trivial to multitask. This is because the Laplacian operator does not couple the components of \mathbf{E} . As a result, we simply set up a task for each component. Note, however that the theoretical overlap is bounded by three rather than four.

Alternitavely, one could divide the grid into subdomains on each level and process them simultaneously. However avoiding data conflicts could be complicated if the scheme is an “update-as-you-go” type rather than completely explicit.

Multitasking the deposition loops is more difficult. We must be careful not to deposit particle quantities from different groups simultaneously. However, we are free to partition each group into tasks because each iteration will assign values to arrays at unique grid cells, as guaranteed by the sorting algorithm.

Even if a routine is parallelized, it still might not achieve significant CPU overlap in a timesharing environment. This area of research is quite new and QN3D has given us insight into some of the issues involved. In particular, we have discovered a relation between the task length (τ_k), the slice (τ_s) and the degree of overlap achieved (\mathcal{M}). Figure 7 shows three possible scenarios.

First note that case 1 illustrates the definition of τ_s . This is the amount of CPU time a code is given each time it is given a processor. τ_s depends upon the size of the code and the priority at which it is run. Case 2 illustrates the definition of the processing time, τ_p . For the section of code illustrated, it is just $\tau_{p1} + \tau_{p2}$. In this case, the CPU time summed over processors, τ_c , would be the sum of the lengths of the non-shaded areas, and $\mathcal{M} = \tau_c / \tau_p$.

Case 1 is the good situation. Here, τ_k is longer than τ_s and we assume that the staircase pattern on the right would repeat several times before the tasks are completed. Case 2 is worse since the first task does not overlap well. Since τ_k is of the order of τ_s , the tasks would complete after one staircase pattern, and the effect of the first part of task 1 on the total overlap will be proportionally larger than the same part in case 1. Case 3 is the worst since most of the work is done before the system knows the code is able to multitask. After a code is given a processor, the system will attempt to give it the next available processor. If the code is not prepared to multitask at that point, then the system will not offer it another processor for the rest of its slice. Thus, if the code starts to multitask in the middle of a slice, long after a second processor became available, additional tasks will be run sequentially until the slice is up.

The actual results will vary since the outcome depends on many factors. But

in QN3D we have different sections of coding that represent the three regimes discussed here that confirm the general hypothesis. The interpolation of the fields to the particles is computationally intensive and when broken into tasks, each task is quite long. It achieves an average overlap of about 3. The particle advance is done for each particle but the calculations are simple so that task length is about the same as the slice. This section gets an overlap of about 2. The deposition of the charges and currents the grid is done in groups, which are frequently small. Breaking up these groups into tasks creates tasks that are shorter than the slice and thus realise almost no overlap. These results could be greatly effected by changes in the operating system's scheduling algorithm.

There are other parameter regimes that affect multitasking performance [24] but these issues here are particularly important in that the results are quite reproducible and the quantities τ_k and τ_s are somewhat controllable, or at least easily measurable to determine the feasibility of multitasking.

3 Single Particle Motion

Before running QN3D to determine electromagnetic fields self-consistently with particle orbits, we followed several particle trajectories in *static* fields. This allowed us to determine the validity of the particle advancement scheme (section 2.3) and to get a feeling for the type of orbits we should expect.

We found it very useful to use a field which could be computed analytically. This gave us the option of either computing the fields on the grid and interpolating to the particle positions or computing the fields at the particle positions directly. These two methods allowed us to separate the numerical effects due to the particle pusher from the effects induced by the interpolation from a finite grid. For the magnetic fields we used the Hill's Vortex. The Hill's vortex is designated by [25],[26]

$$\psi(r, z) = \frac{J_{00}\pi}{2c(1 + \alpha^2)R} r^2 (2R^2 - r^2 - 4\alpha^2 z^2) - \psi_c$$

where

$$\psi = rA_\theta,$$

$J_{0\theta}$ is the o-point current,

R is the o-point radius,

α is a shape factor and

ψ_c is an arbitrary constant that we will set to 0.

From $\psi(r, z)$ we can compute $\mathbf{B}(r, z)$

$$\begin{aligned}\mathbf{B} &= \nabla \times \mathbf{A} \Rightarrow \\ B_r &= \frac{4J_{0\theta}\pi}{c(1+\alpha^2)R} r\alpha^2 z \quad \text{and} \\ B_z &= \frac{2J_{0\theta}\pi}{c(1+\alpha^2)R} (R^2 - r^2 - 2\alpha^2 z^2).\end{aligned}$$

See Fig. 8 for an example of a Hill's Vortex.

We have already demonstrated in section 2.3 that our algorithm conserves energy identically in the case of no electric fields and static magnetic fields. The Hill's vortex simulations confirmed this result.

Since the Hill's vortex is axisymmetric, we would also expect the canonical angular momentum of the particles, P_θ , to be conserved [27], where

$$P_\theta = Mv_\theta r + \frac{e\psi}{c}. \quad (10)$$

However, QN3D does not conserve angular momentum exactly and we did several tests to pinpoint the cause and the severity of the problem.

Though we can not expect the particle pushing scheme to conserve P_θ identically, we did tests to see to how much variation in P_θ was actually due to the pusher itself and how much was due to the finite grid. To do this we compared orbits using fields computed analytically at the particle position and fields interpolated from the computational grid. We found that the orbits resulting from the analytic fields conserved angular momentum to within .2%, while the orbits using the interpolated fields gave a P_θ that varied as much as 30%. This conclusively implicates the finite grid as the cause for significant loss of conservation of P_θ .

However, we have found that the choice of method used to compute P_θ can have an enormous effect on the results. Figure 9 shows P_θ computed from the *same* orbit

(using interpolated fields) but calculated differently. For the left figure we used equation 10 to determine P_θ analytically at the particle position and for the right figure we interpolated ψ from the grid to get P_θ . Not only does the difference in method cause a 12% variation in the result, but even the qualitative form of the temporal variations are almost completely unrelated. This is purely a result of the limits of the diagnostic. Such a problem is unavoidable when the fields have no analytic representation.

In any case we need another criterion to judge the orbits. An interesting method is to use the initial energy and angular momentum to find a surface to which the particle is confined. This energy confining surface (ECS) is independent of time in static fields and thus we can test our particle orbits by how well they are bounded by it. Particles are bounded by

$$r = \pm \frac{P_\theta - \frac{e\psi}{c}}{Mv_0},$$

which is independent of time if $\dot{P}_\theta = 0$. This defines the ECS. Figure 10 shows a particle confined to its ECS for $100\mu s$. Thus we conclude that our particle orbits are reasonable even though we lack conclusive evidence of the conservation of angular momentum.

As an interesting exercise, we followed the magnetic moment of a particle throughout its orbit. Our results were comparable to Schwarzmeier's [8] in that we saw very good conservation along straight field lines, wild fluctuations at the curved ends and a non-adiabatic jump when the particle returned to a straight field line. See Fig. 11

4 Normal Modes

In the single particle motion studies, the magnetic field was static. The next step is to solve the electromagnetic fields on the grid consistently with the particle motion. As a first test of the field-solving algorithm we simulate the normal modes of a plasma with periodic boundaries and a uniform background magnetic field (\mathbf{B}_0) in the z direction. This is essentially a one-dimensional problem and the results are compared to the analytic solutions.

Linear analysis reveals three normal modes of a cold plasma in the Darwin limit.

Two of these are circularly polarized transverse waves propagating parallel to the background magnetic field, and the third is a combination of longitudinal and transverse displacements with propagation perpendicular to the background magnetic field.

To test QN3D, we initialized waves with a given wave number and observed the oscillation with time of the various field quantities. A random point in space was chosen, and the value of the fields at that point were stored and Fourier analyzed in time. The results are presented in Fig. 12 where the error bars are due to the finite Fourier decomposition.

Figure 13 shows B_z Fourier analysed in time for a right-hand mode. The frequency distribution is very peaked showing a definite single frequency of oscillation. We monitored the total energy of the system and even though our field equations are not in conservative form, the total energy is conserved almost perfectly.

The results from the normal mode simulations were very encouraging, indicating the ability of QN3D to model plasma phenomena.

5 The Rigid Rotor Instability

As a second test of QN3D, the rotational instability of the two-dimensional rigid rotor equilibrium is studied. The equilibrium consists of a field-reversed configuration of azimuthal symmetry and infinite length [28]. Ions and electrons have different net rotational frequencies which we will denote as ω_i and ω_e respectively. These rotations provide a net azimuthal current which supports an axial field reversal. This equilibrium is unstable to the rotational instability which has been studied by Harned [29] using a two-dimensional hybrid particle-in-cell code [30]. The rotational instability is a deformation of the plasma cross-section due to its rotation. Fluid theories indicate that the $m = 2$ mode, which is an elliptical deformation, is the most unstable for a field reversed equilibrium [31]. Harned showed that the rotational instability should be less pronounced in larger machines (i.e. larger s). He also has used his code to demonstrate that the rotational instability can be suppressed with the application of quadrupole magnetic fields [32]. Our purpose here is to demonstrate that our code produces results similar to Harned's for the growth rates of the $m = 2$ mode.

The rigid rotor problem adds three complications not encountered in the normal

mode simulations - two spacial dimensions, conducting wall boundaries and a plasma vacuum interface. Solving these problems correctly will suggest that the code may be used to simulate the tilt-mode instability as well.

5.1 Initializing the Rigid Rotor Equilibrium

The particle distribution for a rigid rotor is [29]

$$f(r, \theta, z, v_r, v_\theta, v_z) = \frac{n_0}{(2\pi v_t^2)^{3/2} N} \text{sech}^2\left(\frac{r^2 - r_1^2}{r_0^2}\right) e^{-\frac{v_r^2 + v_z^2}{2v_t^2}} e^{-\frac{(v_\theta - r\omega_i)^2}{2v_t^2}}.$$

Here r_1 is the major radius, r_0 is some measure of the minor radius, N is the total number of particles in the plasma and n_0 is the density at $r = r_1$. Analytically, this particle distribution produces a density profile

$$n_i(r) = n_0 \text{sech}^2\left(\frac{r^2 - r_1^2}{r_0^2}\right),$$

a current profile

$$J_\theta(r) = -ren_i(r)\omega_*$$

and a magnetic field profile [29]

$$B_z(r) = \frac{cM\omega_i^2}{e\omega_*} + \frac{4cMv_t^2}{e\omega_*r_0^2} \tanh[(r^2 - r_1^2)/r_0^2],$$

where $\omega_* \doteq \omega_e - \omega_i$ and

$$v_t \doteq \frac{r_0}{2}(\Omega\omega_* - \omega_i^2)^{1/2}.$$

Ω is the ion cyclotron frequency in the background magnetic field, $B_z(\infty)$. Using Eq. 3 we can compute an electric field as

$$\begin{aligned} E_r(r) &= \frac{1}{en_i c} J_{e\theta} B_z \\ &= -\frac{r\omega_e}{c} B_z. \end{aligned}$$

where ω_e is the electron rotational frequency. Note that this is a *longitudinal* electric field which seems to contradict our assumption that $\nabla \cdot \mathbf{E} = 0$. Actually, this is still consistent to within the order of our approximations as described in the introduction.

One might consider using these analytic expressions to initialize the fields at the beginning of the simulation, but computing fields based solely on particle information provides a much more consistent solution [33]. To do this we use n_i given from the particles to compute \mathbf{J}_e

$$J_{e\theta} = -en_i r \omega_e. \quad (11)$$

ω_e is an input parameter. Next, we introduce the vector potential, \mathbf{A} such that $\nabla \times \mathbf{A} = \mathbf{B}$ and $\nabla \cdot \mathbf{A} = 0$ (the Coulomb Gauge), and apply Ampère's law to get $-\nabla^2 \mathbf{A} = \frac{4\pi}{c} \mathbf{J}$. Given \mathbf{J}_i from the particles and \mathbf{J}_e from Eq. 11 we can solve the above relation for \mathbf{A} . Once we have \mathbf{A} we simply take the curl to get \mathbf{B} . Then we solve

$$\begin{aligned} \mathbf{E} &= \frac{\mathbf{J}_e \times \mathbf{B}}{en_i c} && \text{in the plasma} \\ \text{and} \quad \nabla^2 \mathbf{E} &= 0 && \text{in the vacuum region,} \end{aligned}$$

which gives fields consistent with the initial particle distribution.

5.2 Boundary Conditions

The boundary conditions have been determined by the physical constraints of a conducting wall and assumptions about the symmetry of the problem. To simplify the boundaries we confine the plasma to a box with square, open ends (z boundaries). We expect that the shape of the boundary will have only a limited effect consistent with results found by Harned [33].

On a conducting wall the perpendicular magnetic field must vanish, thus

$$B_x = 0 \quad \text{on the x boundary and}$$

$$B_y = 0 \quad \text{on the y boundary.}$$

Since the equilibrium starts out with only B_z nonzero we assume

$$B_y = 0 \quad \text{on the x boundary and}$$

$$B_x = 0 \quad \text{on the y boundary.}$$

We also assume that the walls are far enough from the plasma so that B_z is essentially uniform, and thus we set

$$\begin{aligned}\frac{\partial B_z}{\partial x} &= 0 \quad \text{on the x boundary and} \\ \frac{\partial B_z}{\partial y} &= 0 \quad \text{on the y boundary.}\end{aligned}$$

For the electric field we have similar restraints. Since there can be no parallel electric field on a perfect conductor we set

$$\begin{aligned}E_y &= E_z = 0 \quad \text{on the x boundary and} \\ E_x &= E_z = 0 \quad \text{on the y boundary.}\end{aligned}$$

By assuming that the entire plasma is charge neutral, we would expect $\nabla \cdot \mathbf{E}$ to be zero on a surface arbitrarily close to the conducting wall, thus we set

$$\begin{aligned}\frac{\partial E_x}{\partial x} &= 0 \quad \text{on the x boundary and} \\ \frac{\partial E_y}{\partial y} &= 0 \quad \text{on the y boundary.}\end{aligned}$$

Particles that contact the conducting wall are lost from the system.

Since the equilibrium is assumed to be uniform in z , we set the axial derivative of all components of all fields equal to zero on the ends of the domain. Particles that leave one end of the device are replaced at the other end as if the domain were periodic in z .

As mentioned previously the explicit field solve iteration scheme is not applicable to low density regions. Instead, we solve $\nabla^2 \mathbf{E} = 0$ in low density regions. It is interesting to note that Harned had trouble with this “two-region” scheme [30]. He observed an anomalous rapid radial diffusion of the particles into the vacuum region by $t = 18\Omega^{-1}$. Thus he constructed a “three-region” scheme which successfully contained the particles. We, however, did not experience any radial diffusion at all out to $t = 142\Omega^{-1}$. Therefore, we have decided to use the two-region scheme as long as it seems adequate.

5.3 Results

To monitor the growth of the rotational instability, we did as Harned [29] and Fourier analysed δr in theta and plotted the azimuthal mode number, m . The dominant mode was $m = 2$ (Fig. 14). One can observe the qualitative formation of the rotational instability by plotting the particle currents or positions in the x-y plane (Fig. 15), or by plotting the particle distribution in theta (Fig. 16). We did several runs for different values of $\alpha \equiv -\omega_i/(\omega_e - \omega_i)$. Our results basically agree with Harned's but differ significantly for low α as shown in Fig. 17.

The origin of this discrepancy is not yet understood. Further investigation seems to indicate that particles near the plasma-vacuum interface increase in angular momentum giving the plasma an effectively higher α than the initial conditions. This, in-turn, causes the growth of the rotational instability.

For the low α cases, we investigated the numerical effects of increasing the number of particles and the number of grid cells. Neither change effected the results of the simulations. This is fortunate for the computational model in general since it demonstrates the code's insensitivity to these parameters. But, unfortunately for the rotation simulations, increasing the number of particles or the number of grid cells did not improve the low α results. However, the three-dimensional equilibria used for the tilt mode problems of the next section did not exhibit the rotational instability even though the ions had no rotation (i.e. low α). Thus we conclude that though this problem is troublesome, apparently, it does not effect the tilt mode results.

In addition, we also monitored the system energy as was done in the normal mode simulation. Again, we find that the total energy is conserved very well. We also monitored the magnetic flux through an arbitrary plane perpendicular to the z axis, and found that it was conserved very well as we would expect in the presence of a conducting boundary. The parameters for the rigid rotor runs are given in Table 2.

6 Tilt Mode Problem

6.1 Initializing the FRC Equilibrium

The FRC equilibrium configuration is assumed to be axisymmetric and symmetric about the midplane. QN3D accepts as input, the axisymmetric magnetic field, the ion particle density and the ion temperature on a two-dimensional r - z grid which extends from the midplane outward and from the axis outward. As a result of these symmetries, the code computing the fields must only supply them on one fourth of the r - z plane. These quantities are mapped to the rest of the three dimensional domain using these symmetries. The runs done here use equilibria computed by EQV, a kinetic equilibrium code developed by Shumaker [34].

6.2 Circular Boundaries

A further refinement of the boundaries was added for the FRC problem. The basic idea is to model the cylindrical wall as closely as possible. Even though we mentioned that it was not necessary for the rigid rotor problem, it is necessary here if we want to include passive mirrors in the machine.

To model this configuration, we assume that any grid point (I,J,K) is part of the computational region if and only if

$$XG(I)**2 + YG(J)**2 \leq RG(K)**2$$

where $RG(K)$ is the radius of the perfectly conducting wall at $ZG(K)$. To implement this condition we label any point exterior to the coil as a vacuum cell. Thus it is ignored by the plasma field solve. In the vacuum, we assume that the electric field is zero outside the coil therefore $\nabla \times \mathbf{E} = 0$ and thus $\dot{\mathbf{B}} = 0$.

6.3 Observing the Tilt Mode

As mentioned in the introduction, we expect the tilt mode to be unstable when the ion gyroradius becomes small with respect to the size of the plasma. A convenient measure of this relative size is s , which, itself, is a measure of the number of ion gyroradii between the o-point and separatrix. For higher s we would expect the plasma to act like a fluid. Analytically [35]

$$s \doteq \int_R^{r_s} \frac{r}{r_s \rho_i(r)} dr, \quad (12)$$

where R is the o-point radius, r_s is the separatrix radius and ρ_i is the ion gyroradius. We investigated two cases, one with $s = 1.6$ and another with $s = 12$. The parameters of these runs are summarized in Table 3. These two equilibria have similar sizes and the same ion temperature. Thus to increase s the magnetic field must be increased. This leads to an increase in the magnetic field pressure, which requires a higher density. Alternatively, s could have been increased by increasing the size of the plasma.

We used many diagnostics to observe the tilt mode. First, we plotted flux contours. For a cylindrically symmetric configuration the flux contours are also contours of constant ψ ($\doteq r A_\theta$) and we find A_θ by solving

$$-\nabla^2 \mathbf{A} = \frac{4\pi}{c} \mathbf{J}.$$

This was very helpful in that we were able to see the plasma tilting very easily in the high- s case and not at all in the low- s case. However, when the plasma starts to tilt, it loses its axisymmetry and the contours of constant ψ are no longer the flux surfaces of the magnetic field. Nevertheless, these plots gave us a qualitative indication that the phenomena was occurring in the high- s regime and not in the low- s regime (see Figs. 18 & 19).

For another visual diagnostic we simply plotted contours of constant particle density. This showed very clearly that one case tilted while the other did not (see Figs. 20 & 21).

It is important to remember that the tilt mode instability observed here grew out of the noise in the simulation introduced by the random nature of the particle initialization. No initial perturbation was employed to help the plasma develop the tilt.

For a quantitative diagnostic, we tried a method suggested by Tuszewski [36] of the Los Alamos National Laboratory. He suggested that we simulate the measurement of the Faraday rotation of a light beam shot through the plasma. This has special

relevance for him since it is a measurement he can do in the laboratory. This rotation is proportional to the integral of the density multiplied by the normal magnetic field [37], i.e.

$$\Theta_F \propto \int n_i \mathbf{B} \cdot d\mathbf{l}, \quad (13)$$

where the integral is along the beam. From symmetry, it is clear that Θ_F is initially zero in an FRC. However, as the tilt mode develops, some rotation should be noticeable (Fig. 22). In fact, we should be able to recognize the tilt mode by the signature in a plot of Θ_F as a function of z (Fig. 23).

Implementing this diagnostic was trivial. We simply did the integral of Eq. 13 along several grid lines in the x - z plane to find Θ_F as a function of z . We also did it along lines in the y - z plane in case the plasma tilted in that plane.

By plotting Θ_F as a function of z we saw the tilt signature clearly in the high- s case but it was absent in the low- s case (Fig. 24). In order to pick the tilt mode signature out of the noise we fitted the data to a polynomial with the same signature. In particular, we found the least-squares fit to

$$f_x(z) = (Az^4 - Bz^8)e^{-(z/L)^4}$$

where L is predetermined and A and B are found by the fitting procedure. The x subscript indicates that the polynomial fits the data found from the Faraday rotation diagnostic done in the x - z plane. A similar function, $f_y(z)$, was found for the y - z plane data. To get a magnitude from these functions we simply integrated the square of these functions, took the sum and then the root, i.e.

$$|\Theta_F| \doteq \left[\int_{zmin}^{zmax} (f_x^2(z) + f_y^2(z)) dz \right]^{1/2}.$$

Using both f_x and f_y allows us to observe the tilt mode if it does not happen to occur in either of the planes completely. $|\Theta_F|$ is recorded as a function of time for both cases and growth rates are computed. The results are quite clear. The high- s case shows the tilt growing with a growth rate close to MHD predictions. The low- s

case shows only slight growth if any at all. Other polynomials with the same basic structure were used to fit the simulated data. They all gave similar results.

These results compare favorably with preliminary results from Barnes, et al. [9] (see Fig. 25). Their results were normalized to the MHD growth rates. There is some ambiguity as to the definition of the MHD growth rate [1][2]. We obtained the best consistent results with

$$\gamma_{MHD} = 2 \frac{v_A}{L},$$

where L is the axial length of the plasma. This is the inverse of the time it takes for an Alfvén wave to traverse the system. We would not expect exact agreement since their equilibria are different and they are measuring the displacement of the flux surfaces rather than the Faraday rotation. However, the general agreement is very encouraging.

As usual, we also recorded the energy of the system as a function of time for each case. Both cases conserved energy to within 1 percent.

7 Conclusions

Quite fortunately, we have been able to give strong credence to the initial hypothesis that the tilt mode will exist in regimes of higher s . This result, by itself, is important for those planning to build larger FRC experiments. But, in addition, QN3D has a major advantage in that it should be able to model the nonlinear regime of the tilt mode which will be even more crucial to the future of FRC experiments.

In the process of investigating the tilt mode, we have developed a major computational tool which should continue to provide important results to many other plasma physics problems that could benefit from particle simulations.

It is worthwhile to note that the computations presented here would have been impossible without the Cray-2 computer. The field length of QN3D for these runs was about 25 million words (which is equivalent to 1.6 trillion bits). The largest Cray X-MP computers to date have only 16 million words of memory. Each run of the tilt mode problem used 50 hours of Cray-2 resource time and took nearly a week to

complete in the time-sharing environment. Though they are still quite expensive, as computers continue to get larger and faster, particle codes will become more popular.

ACKNOWLEDGMENTS

This project, in pushing algorithms and computer systems to their limits brought forth problems that needed the personal attention of many specialized talents. Thus we would like to acknowledge those who gave generously of their time, resources and expertise to help me complete my research.

We wish to acknowledge the assistance of several other outstanding scientists who offered invaluable advice and support throughout the course of my research. In particular we thank Dan Barnes, John Bolstad, Bruce Cohen, Alex Freidman, Dennis Hewett, Douglas Harned and Tony Sgro – each for much time spent and much knowledge conveyed.

We are also grateful for the assistance of the entire staff of the NMFECC. In particular – to Larry Berdahl for assistance with vectorization; to Jean Shuler for assistance with graphics; and to Dave Storch, Michael Ganzberger and the rest of the Systems Group for helping to sort out a host of systems problems.

Supplemental computer resources were generously provided by the Controlled Thermonuclear Research Division of the Los Alamos National Laboratory.

This work was performed under the auspices of the United States Department of Energy by the Lawrence Livermore National Laboratory under contract W-7405-ENG-48.

References

- [1] R.E. Siemon, W.T. Armstrong, D.C. Barnes, R.R. Bartsch, R.E. Chrien, J.C. Cochrane, W. Hagrass, R.W. Kewish, P.L. Klinger, N.R. Lewis, R.K. Linford, K.F. McKenna, R.D. Milroy, D.J. Rej, J.L. Schwarzmeier, C.E. Seyler, E.G. Sherwood, R.L. Spencer and M. Tuszewski, *Fusion Technology* **9**, 13 (1986).
- [2] J.L. Schwarzmeier, D.C. Barnes, D.W. Hewett, C.E. Seyler, A.I. Shestakov and R.L. Spencer, *Phys. Fluids* **26**, 1295 (1983).
- [3] J.N. Leboeuf, T. Tajima and J.M. Dawson, *J. Comput. Phys.* **31**, 379 (1979).
- [4] F. Brunel, J.N. Leboeuf, T. Tajima, J.M. Dawson, M. Makino and T. Kamimura, *J. Comput. Phys.* **43**, 268 (1981).
- [5] F. Brunel, T. Tajima, *Phys. Fluids* **26**, 535 (1983).
- [6] D.C. Barnes, A.Y. Aydemir, D.V. Anderson, A.I. Shestakov and D.D. Schnack, Proceedings of the Third Symposium on the Physics and Technology of Compact Toroids in the Magnetic Fusion Energy Program (1980) 134.
- [7] C.E. Seyler and D.C. Barnes, *Phys. Fluids* **24**, 1989 (1981).
- [8] J.L. Schwarzmeier and C.E. Seyler, *Phys. Fluids* **27**, 2151 (1984).
- [9] D.C. Barnes, J.L. Schwarzmeier, H.R. Lewis and C.E. Seyler, *Phys. Fluids* **29**, 2616 (1986).
- [10] J.P. Freidberg, *Phys. Fluids* **15**, 1102 (1972).
- [11] J.S. Kim, Ph.D. Thesis, UC Berkeley, 1984.
- [12] K.R. Symon, C.E. Seyler and H.R. Lewis, *J. Plasma Phys.* **27 part 1** (1982) 13.
- [13] H.R. Lewis, D.C. Barnes, J.L. Schwarzmeier, C.E. Seyler and K.R. Symon, *Phys. Fluids* **28**, 3546 (1985).
- [14] A.N. Kaufman and P.S. Rostler, *Phys. Fluids* **14**, 446 (1971).

- [15] E.J. Horowitz, Ph.D. Thesis, UC Davis, 1987.
- [16] D.R. Nicholson, Introduction to Plasma Theory, John Wiley & Sons, New York, 1983.
- [17] D. Hewett, private communication.
- [18] A. Brandt, *Mathematics of Computation* **31**, 333 (1977).
- [19] K. Stüben and U. Trottenberg, Lecture Notes in Mathematics **960**, Springer-Verlag, Berlin 1981.
- [20] A. Brandt, Lecture Notes in Mathematics **960**, Springer-Verlag, N.Y. 220 (1982).
- [21] E.J. Horowitz, *J. Comp. Phys.*, **68**, 56 (1987).
- [22] C.K. Birdsall and A.B. Langdon, Plasma Physics via Computer Simulation, McGraw-Hill, New York, 1985.
- [23] D.V. Anderson, NMFECC *Buffer* **10**, 10 (Oct. 1986).
- [24] NMFECC Computational Physics Group, Bull. Am. Phy. Soc. **31**, 8T24-25, 1986.
- [25] V.D. Shafranov, Reviews of Plasma Physics, **2**, Consultants Bureau, New York, 1966 p1607.
- [26] D.E. Shumaker, Ph.D. Thesis, UC Davis, 1977.
- [27] G. Schmidt, Physics of High Temperature Plasmas, 2nd ed., Academic Press, New York, 1979.
- [28] R.L. Morse and J.P. Freidberg, *Phys. Fluids* **13**, 531 (1970).
- [29] D.S. Harned, *Phys. Fluids* **26**, 1320 (1983).
- [30] D.S. Harned, *J. Comput. Phys.* **47**, 452 (1982).
- [31] C.E. Seyler, *Phys. Fluids* **22**, 2324 (1979).

- [32] D.S. Harned, *Phys. Fluids* **27**, 554 (1984).
- [33] D.S. Harned, private communication.
- [34] D.E. Shumaker, Bull. Am. Phy. Soc. **31**, 1986.
- [35] J.T. Slough, A.L. Hoffman, R.D. Milroy, D.G. Harding and L.C. Steinhauer, *Nuclear Fusion* **24**, 1537 (1984).
- [36] M. Tuszewski, private communication.
- [37] N.A. Krall and A.W. Trivelpiece, Principles of Plasma Physics, McGraw-Hill, New York 1973.

Table Captions

1: Parameter Regime for Normal Mode Analysis: Values in the right-most column are for the perpendicular mode when different from the other two modes.

2: Parameter Regime for Rigid Rotor Problem: Some values varied from run to run.

3: Tilt Mode Parameters: Both cases used 1,000,000 particles in a grid 41 cubed.

TIME STEPS	1000	
PARTICLES	400000	
TIME STEP	1.262×10^{-8} sec	
ION DENSITY	10^{15} particles/cc	
CUT-OFF DENSITY	10^{14} particles/cc	
IONS/PARTICLE	4.16×10^{12}	10^{14}
DOMAIN SIZE	$20\text{cm} \times 16\text{cm} \times 42\text{cm}$	$20\text{cm} \times 16\text{cm} \times 125\text{cm}$
GRID SIZE	$12 \times 17 \times 49$	
GRID CELL SIZE	$1\text{cm} \times 1\text{cm} \times .87\text{cm}$	$1\text{cm} \times 1\text{cm} \times 2.6\text{cm}$
PARTICLES/GRID CELL	26	
ION CYCLOTRON FREQUENCY	$3.86 \times 10^7 \text{sec}^{-1}$	
BACKGROUND MAGNETIC FIELD	4×10^3 Gauss	
ELECTRIC FIELD PERTURBATION	1×10^{-3} Statvolt/cm	

Table 1:

TIME STEPS	1400
PARTICLES	200000
TIME STEP	2×10^{-9} sec
ION DENSITY	10^{15} particles/cc
CUT-OFF DENSITY	10^{14} particles/cc
IONS/PARTICLE	4.7×10^{12}
DOMAIN SIZE	48.7cm \times 48.7cm \times 87.4cm
GRID SIZE	39 \times 39 \times 7
GRID CELL SIZE	1.35cm \times 1.35cm \times 2.19cm
PARTICLES/GRID CELL	23
ION CYCLOTRON FREQUENCY	$4 - 8 \times 10^7 \text{sec}^{-1}$
BACKGROUND MAGNETIC FIELD	$4 - 8 \times 10^3$ Gauss
MAJOR RADIUS	4.06cm
MINOR RADIUS	4.49cm

Table 2:

	$s = 12$	$s = 1.6$
Background B	$2.2 \times 10^4 \text{ G}$	$2.7 \times 10^3 \text{ G}$
Peak Density	$7.0 \times 10^{16}/\text{cm}^3$	$1.1 \times 10^{15}/\text{cm}^3$
Alfvén Speed	$1.82 \times 10^7 \text{ cm/s}$	$1.78 \times 10^7 \text{ cm/s}$
Ion Temperature	200 eV	200 eV
Separatrix Length	33 cm	35 cm
Separatrix Radius	12 cm	12 cm
Wall Radius	20 cm	20 cm
MHD Growth Time	$0.97 \mu\text{s}$	$0.99 \mu\text{s}$
Computed Growth Time	$0.92 \mu\text{s}$	$12.6 \mu\text{s}$
Poloidal Flux		
Encircled by o-point radius	$2.0 \times 10^6 \text{ G} \cdot \text{cm}^2$	$2.5 \times 10^5 \text{ G} \cdot \text{cm}^2$
Enclosed by wall	$1.8 \times 10^6 \text{ G} \cdot \text{cm}^2$	$2.2 \times 10^6 \text{ G} \cdot \text{cm}^2$

Table 3:

Figure Captions

1: Example of the tilt mode in an FRC: Plotted is A_θ . This is the result of QN3D advancing the equilibrium for about one microsecond.

2: The Computational Cycle: The code starts with fields given on the grid at time step 0, particle positions given at time step 0 and particle velocities given at time step -1/2. The cycle starts with interpolating fields to the particle positions

3: Multigrid – Cycle C Flow Chart: This version differs slightly from Brandt's [18] in that we do not bother to check for convergence at each grid level.

4: Multigrid and the Plasma-Vacuum Interface U is plotted versus x . The plasma field is assumed to be given and the vertical lines delimit the boundaries between the plasma and the vacuum. On the coarser grids denoted by circles, the boundary value is given by the first circles within the plasma region. These values are greatly different than the boundary values of the finer grid.

5: Bi-Linear Weighting: The "x" marks the center of the particle.

6: Sorted Particles in the Grid

7: The Relation Between τ_k and τ_s : Each horizontal bar represents a CPU. The shaded regions are other users in the time-sharing environment. ST denotes single threading code and T1, T2... denote tasks 1 through 4.

8: Hill's Vortex: Contours of constant ψ : $R = 8\text{cm}$, $\alpha = .111$ and $J_{0\theta} = 10^{13}$ statamps/cm²

9: Grid Effects on Computing P_θ : The canonical angular momentum is plotted as a function of time normalized to its initial value. Note: both plots are from the same orbit.

Left: Analytically computed ψ – Right: Interpolated ψ .

10: Particle Orbit in the ECS: The particle's orbit is traced in the r - z plane. Top: Particle orbit for 500 steps showing clearly the ECS and also showing the particle bouncing off the end of the surface. – Bottom: Same orbit for 10,000 steps showing many transits of and bounces in the ECS. The plotting resolution is much coarser for the bottom plot making the particle appear to move in straight line segments. Note that 10,000 is much longer than any currently planned run.

11: Magnetic Moment versus Time: The magnetic moment is normalized to its initial value. This orbit used fields interpolated from the grid, but similar results were found from orbits using analytic fields.

12: Frequency versus Wave Number

13: Fourier analysis of B_x in time

14: $m = 2$ as the Dominant Mode: For $\alpha = 1.$, the amplitude is shown for $m = 0, \dots, 5$ versus time.

15: Observing Growth of Rotational Instability: Ion current and particle positions in x-y plane. $t = 0$ and $t = 142 \Omega^{-1}$

16: Particle Distribution in θ Showing Growth of Rigid Rotor Instability: Distribution of particles in θ . Top row: $t = 0$, Bottom row: $t = 142 \Omega^{-1}$.

17: Harned's Results - γ/Ω versus α : This figure is from the Physics of Fluids **26**, May 1983, page 1322. The stars indicate our results. Reproduced with permission of the author and the American Institute of Physics. ©Copyright AIP 1983.

18: Tilting ψ Surfaces: $s = 12$: Top: $0\mu s$ Bottom: $2\mu s$

19: Non-Tilting ψ Surfaces: $s = 1.6$: Top: $0\mu s$ Bottom: $2\mu s$

20: Tilting Density Surfaces: $s = 12$: Top: $0\mu s$ Bottom: $3\mu s$

21: Non-Tilting Density Surfaces: $s = 1.6$: Top: $0\mu s$ Bottom: $3\mu s$

22: Faraday Rotation Concept: Depicted is one magnetic field surface from the midplane outward. The vertical lines represent light beams from a diagnostic device. The segments colinear with the beams represent the contribution of the field to Θ_F . In the top figure all of the contributions cancel at all axial positions. In the lower figure, a tilt has developed and the contributions no longer cancel. Note that for the inner beams Θ_F is positive while for the outer beams, it is negative.

23: Expected Tilt Mode Signature from Faraday Rotation Diagnostic: Θ_F is plotted as a function of z . Cases that tilt should show this qualitative structure.

24: Faraday Rotation Results: The top figure is for $s=12$ and the lower figure is for $s=1.6$. The smooth lines marked with an 'f' are the fitted polynomials.

25: Barnes' Results: This figure is from the Physics of Fluids **29** page 2623, August 1986. The stars indicate our results. Reproduced with permission of the authors and the American Institute of Physics. ©Copyright AIP 1986.

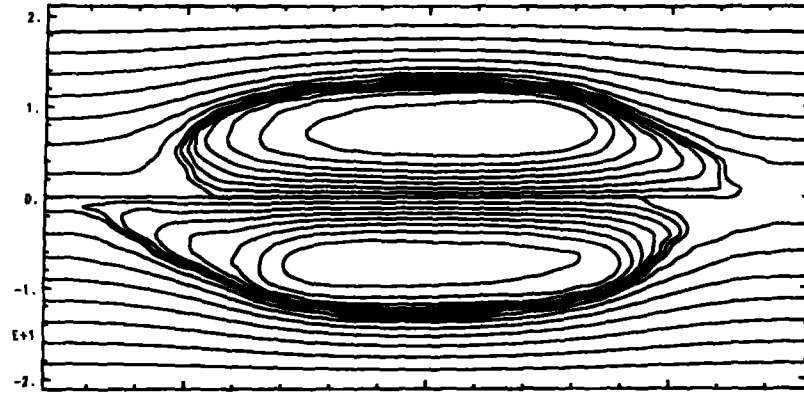


Figure 1:

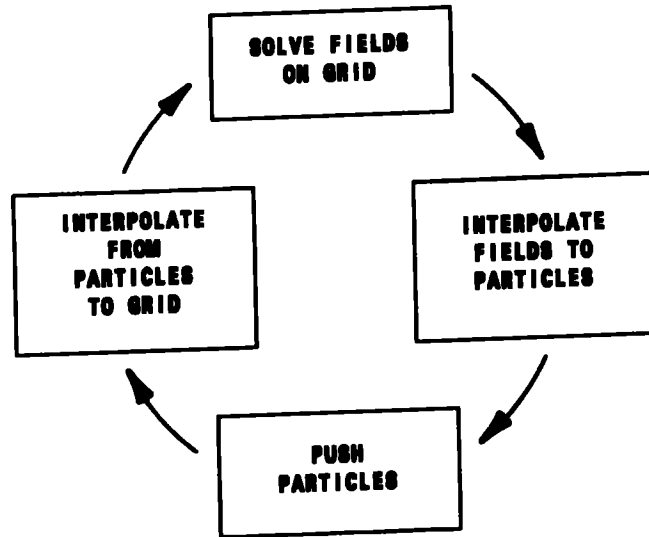


Figure 2:

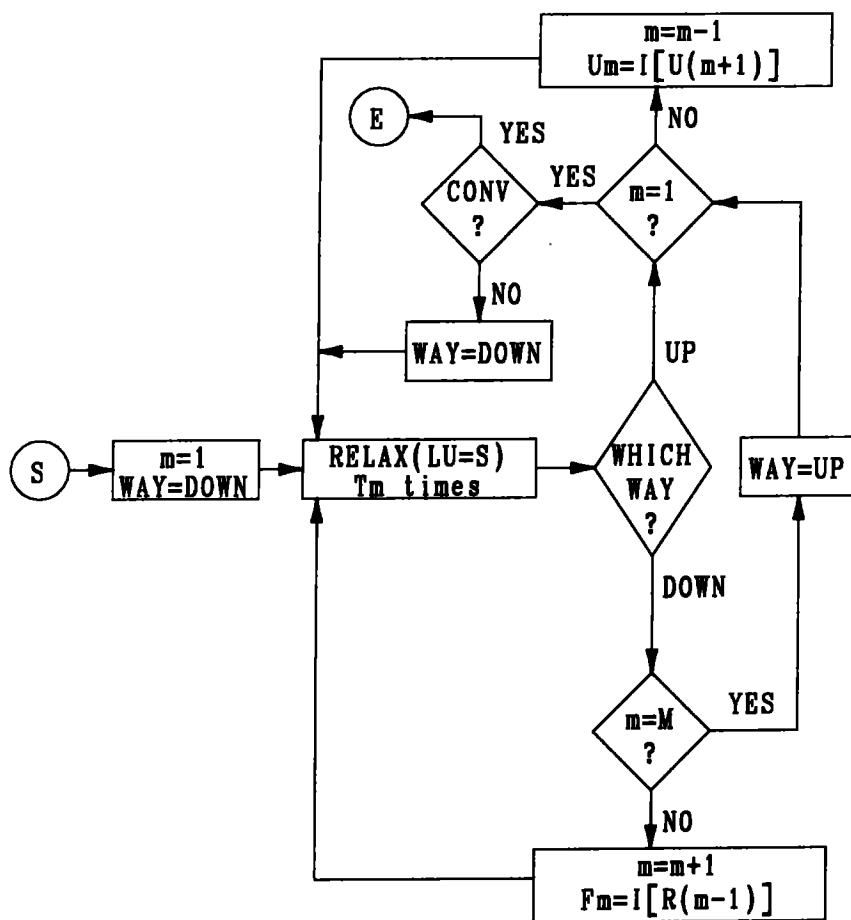


Figure 3:

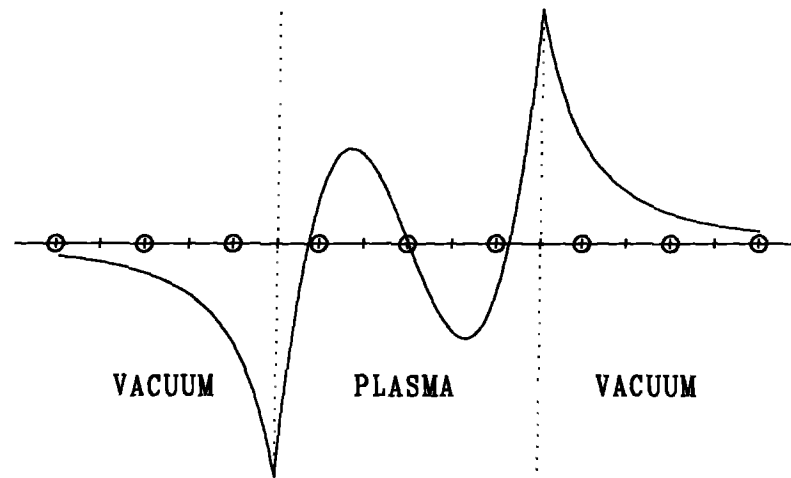


Figure 4:

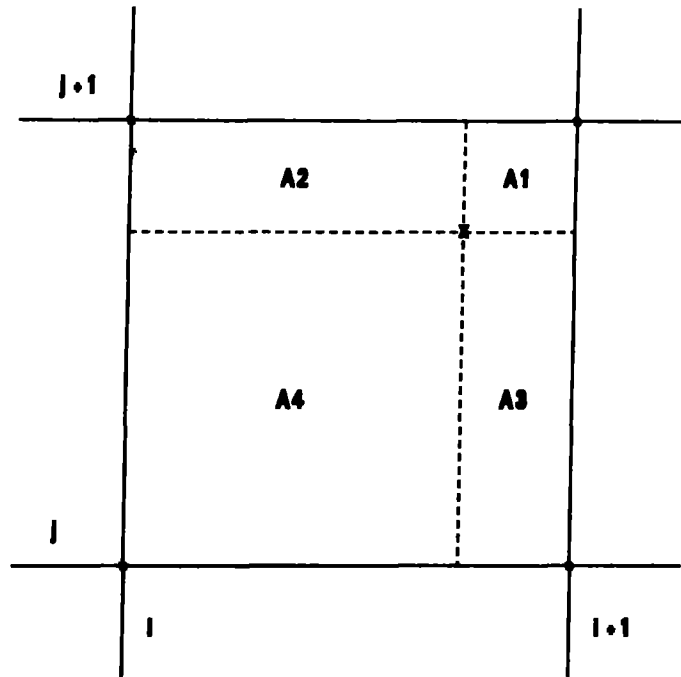


Figure 5:

2 1	2 1 4	1 2	1 2
1	1 3 2	1 5 3 4	2 2 3 1
2 1	4 1 5 2	2 3	2 1
2 1	2 1	3 1	1 2

Figure 6:



Figure 7:

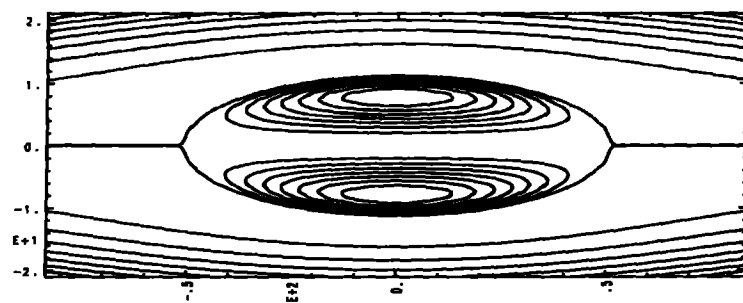


Figure 8:

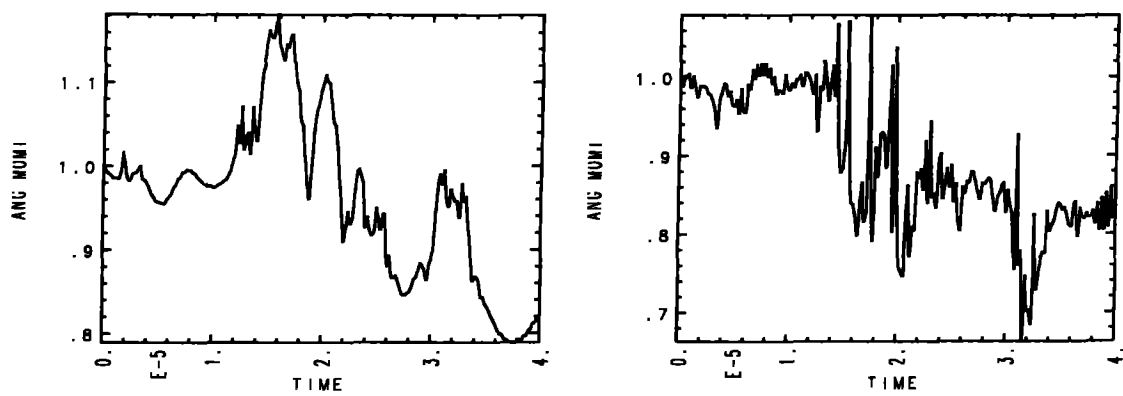


Figure 9:

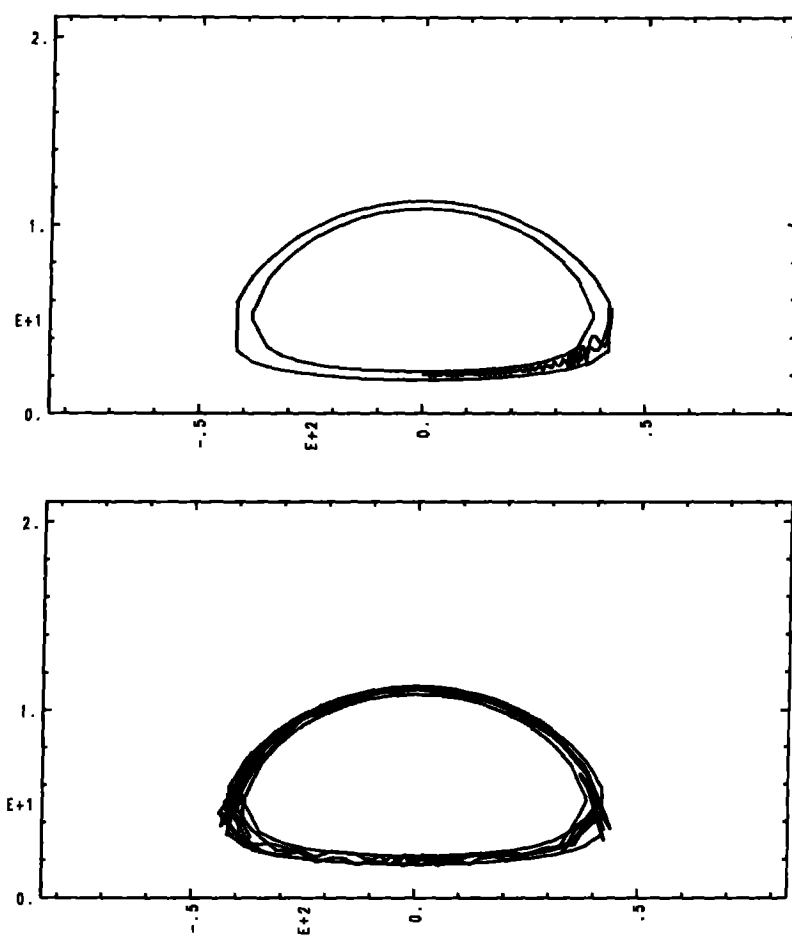


Figure 10:

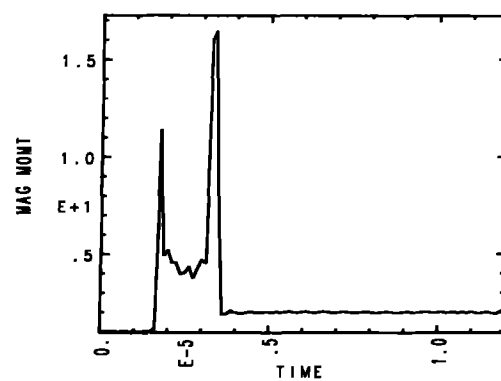


Figure 11:

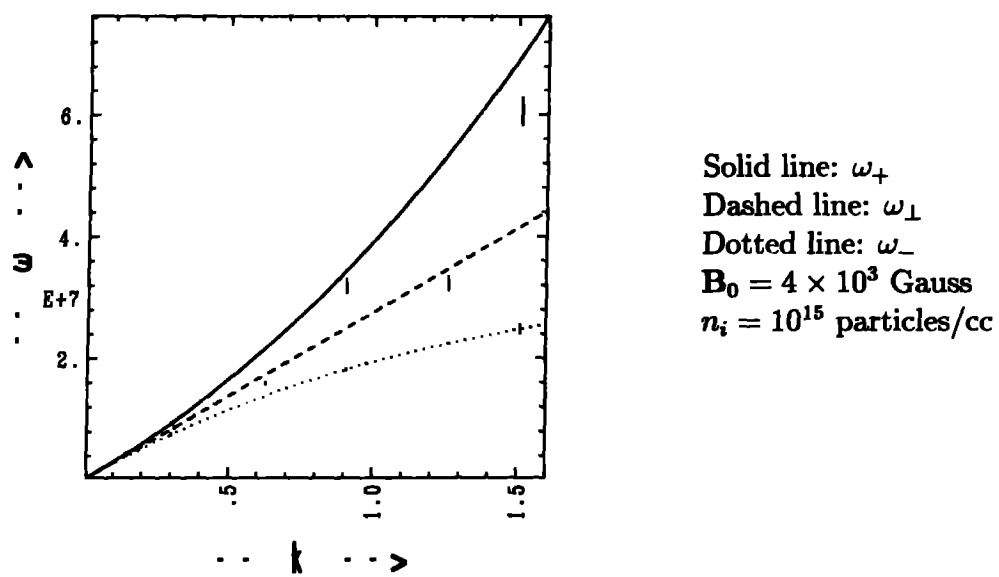


Figure 12:

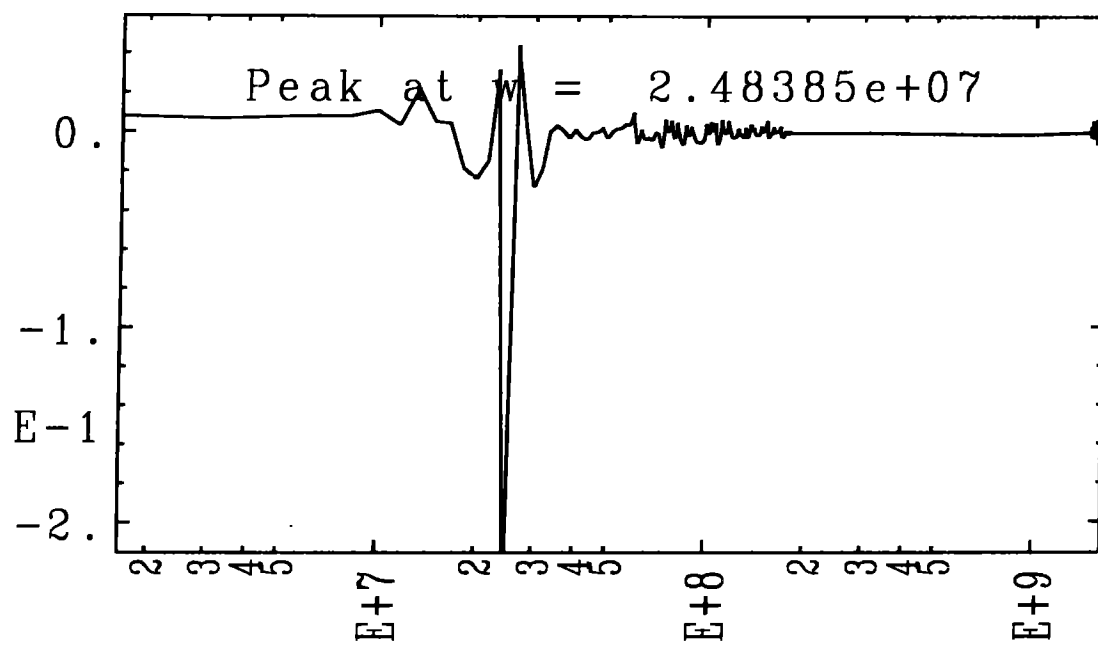


Figure 13:

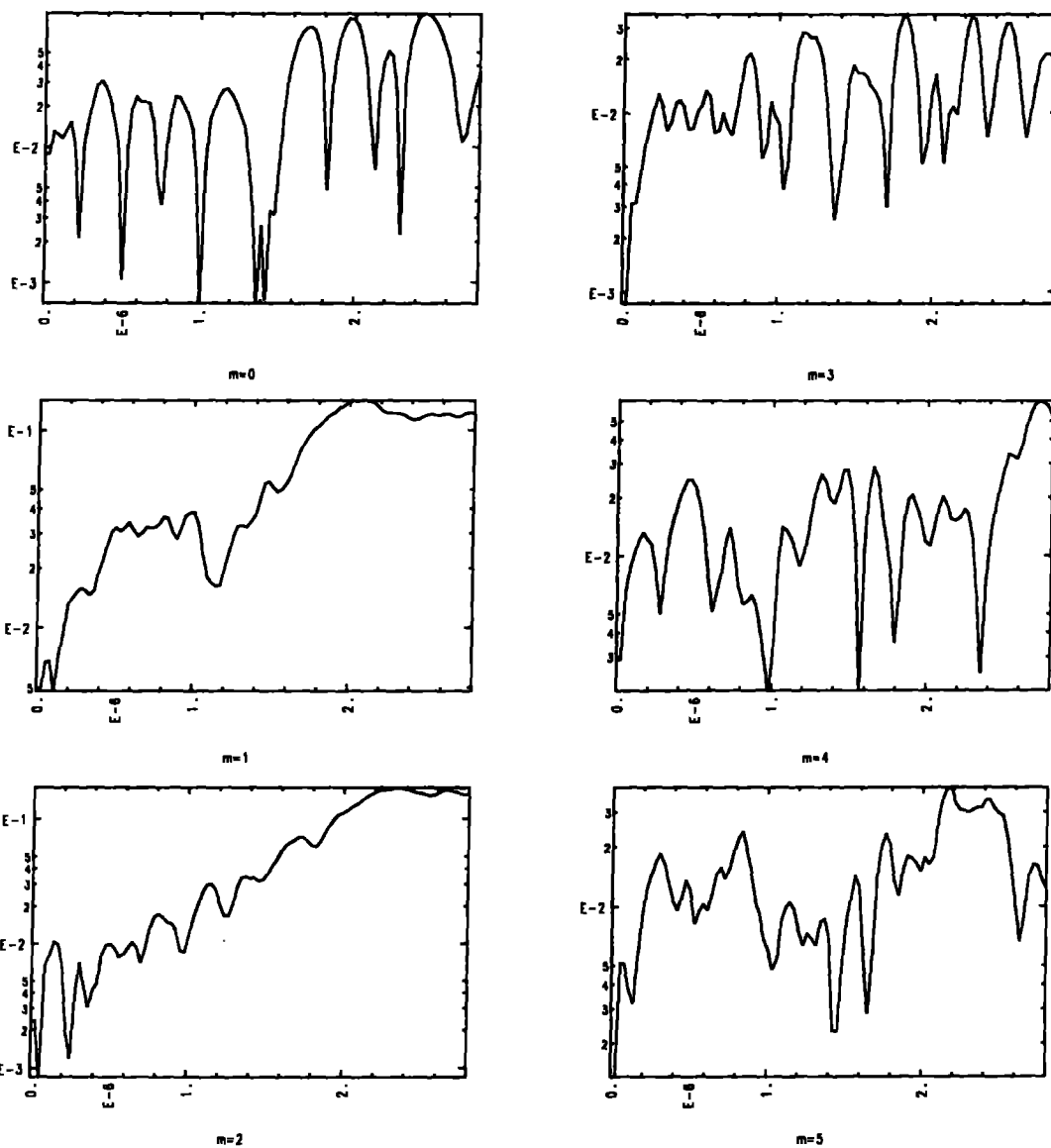


Figure 14:

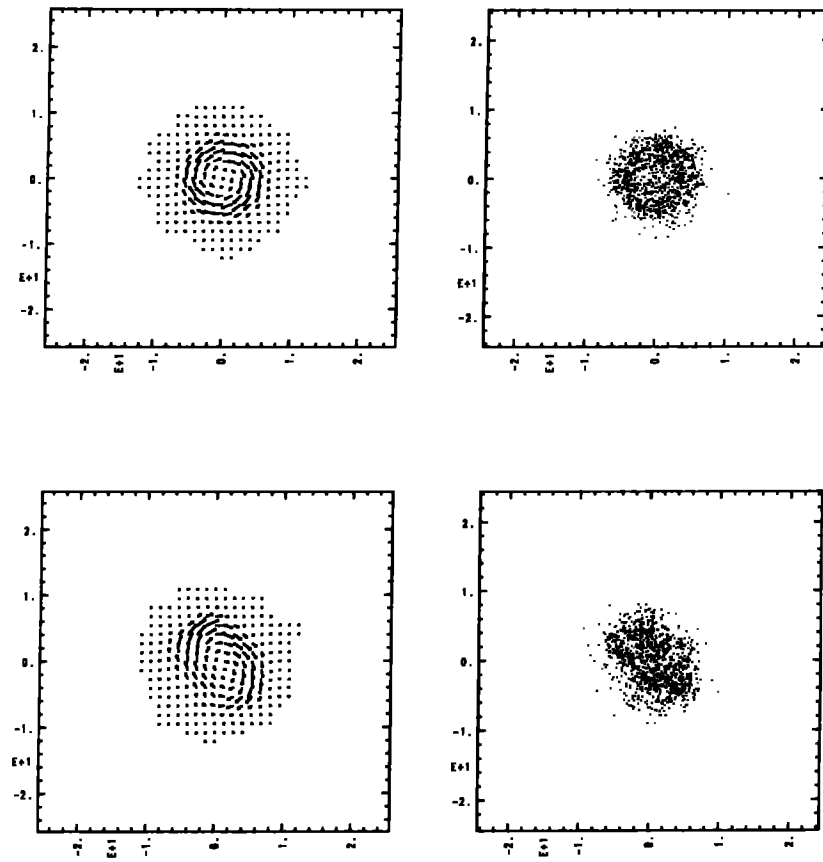


Figure 15:

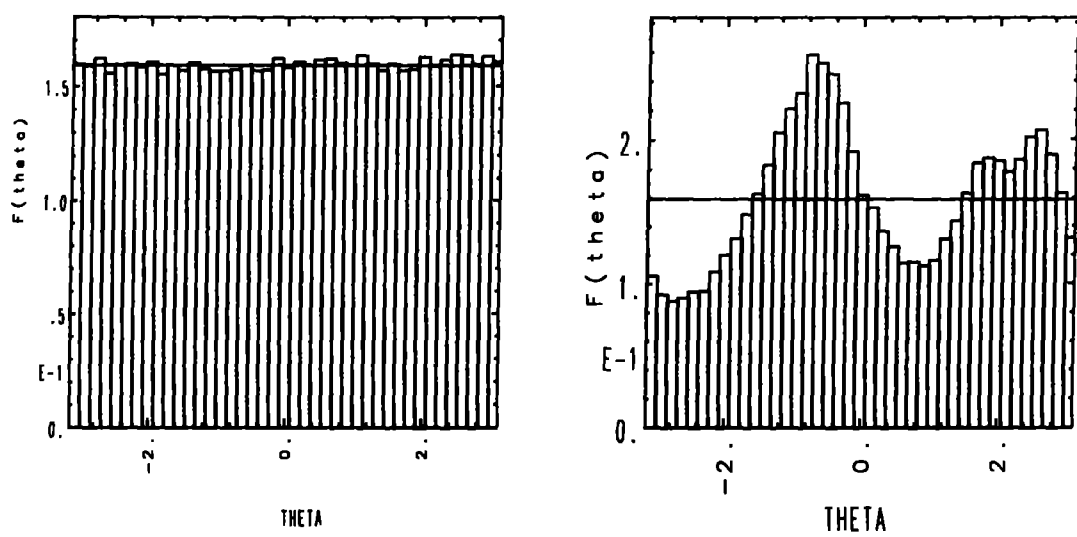


Figure 16:

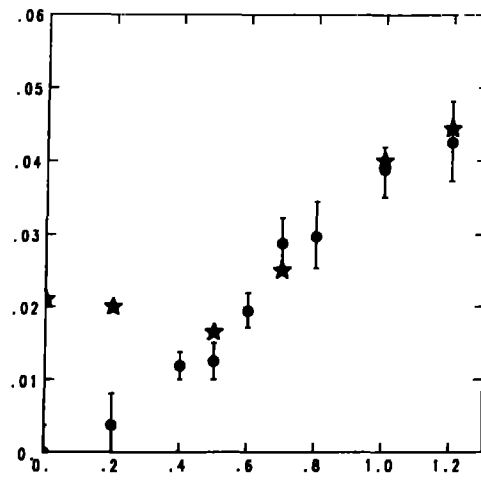


Figure 17:

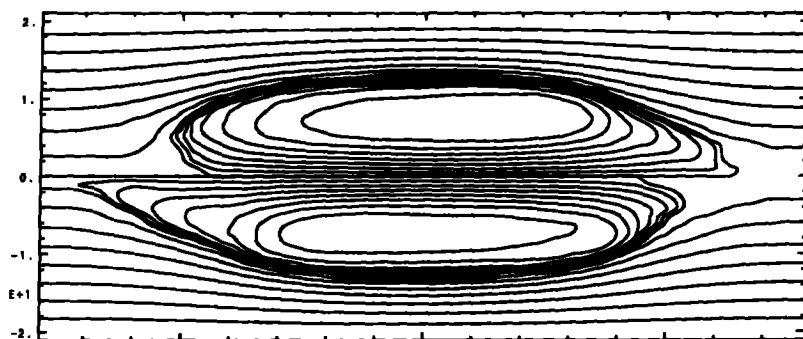
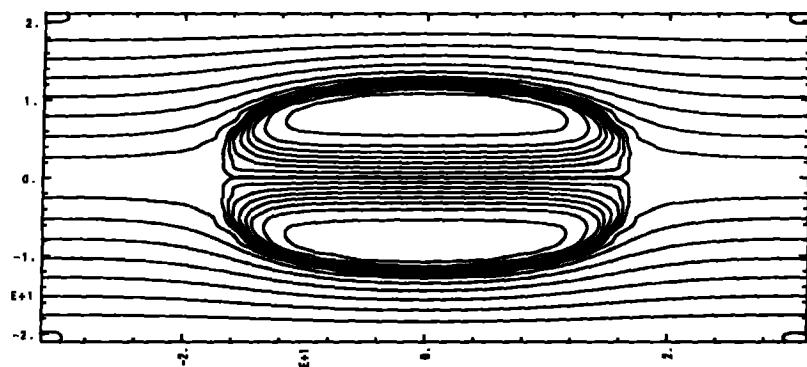


Figure 18:

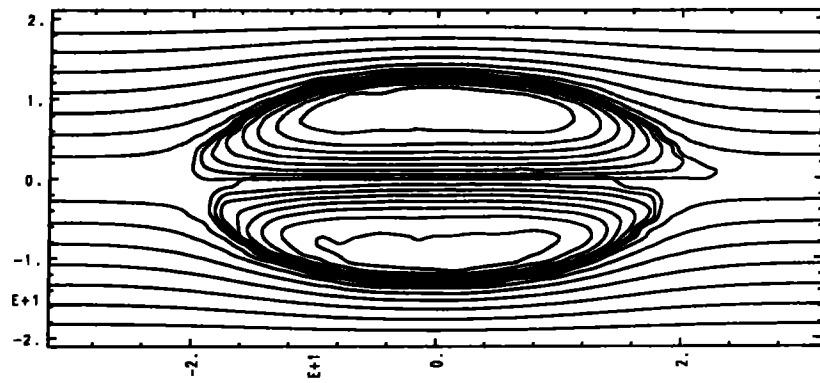
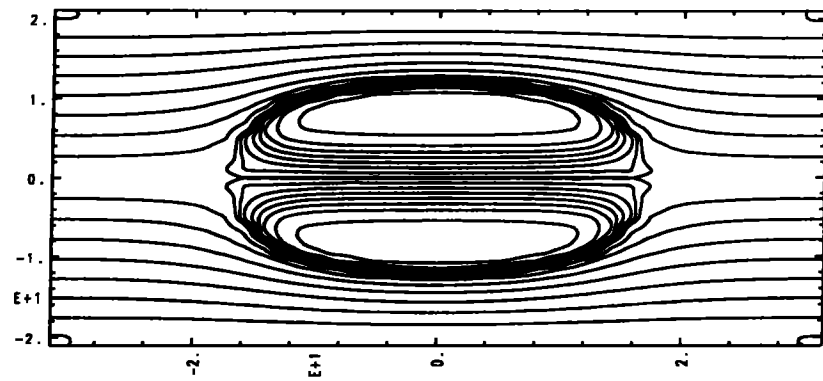


Figure 19:

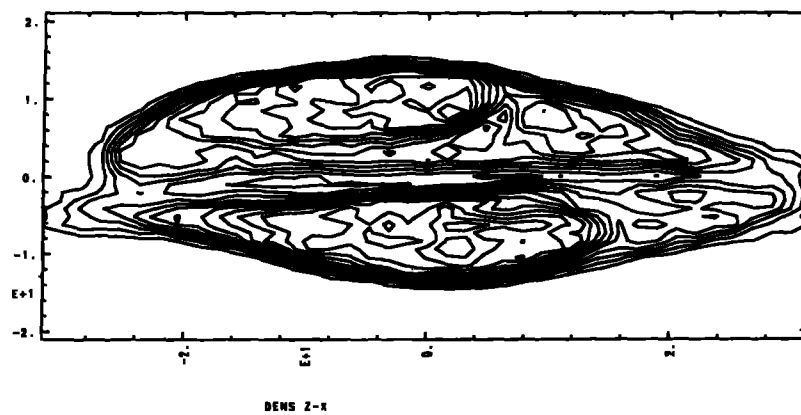
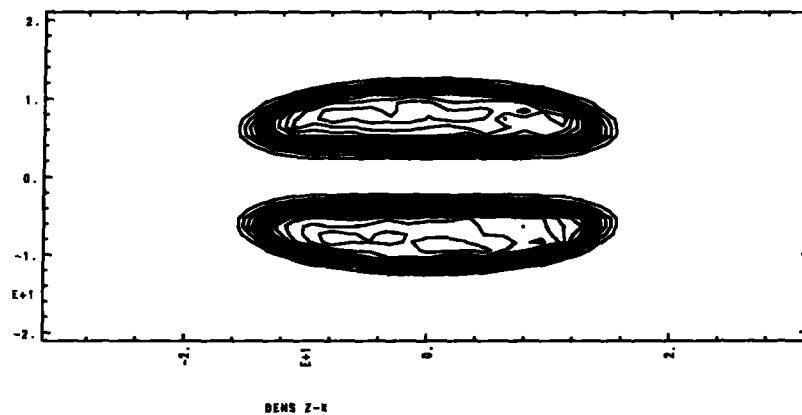


Figure 20:

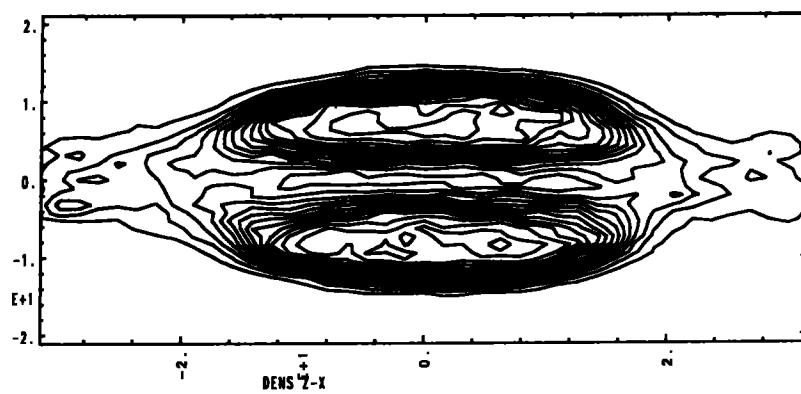
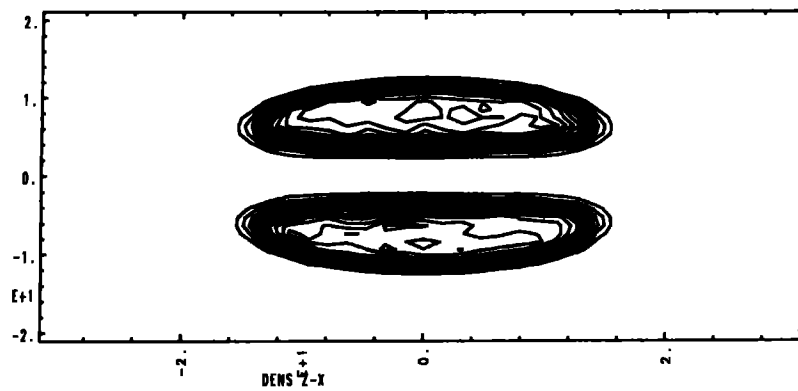


Figure 21:

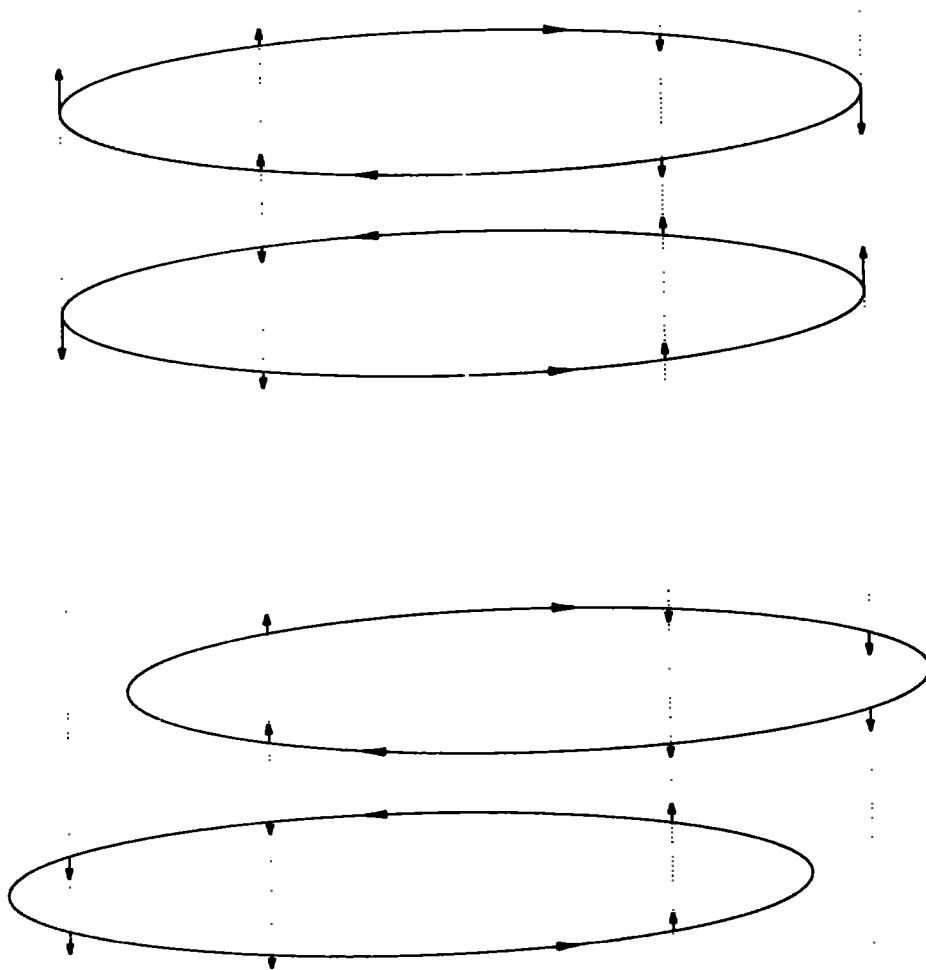


Figure 22:

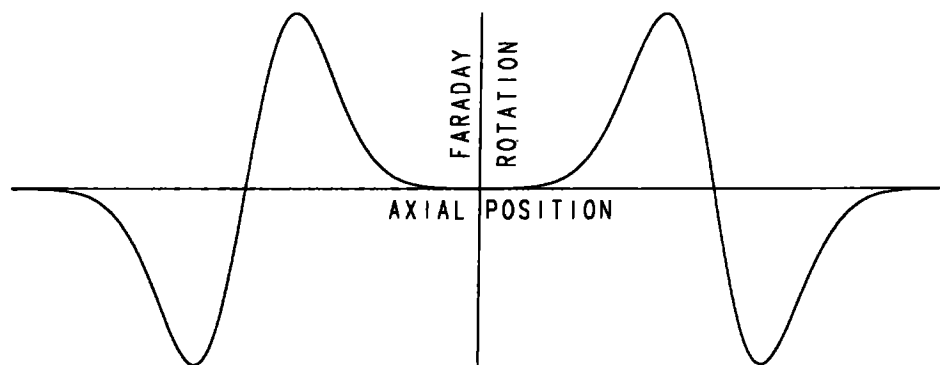


Figure 23:

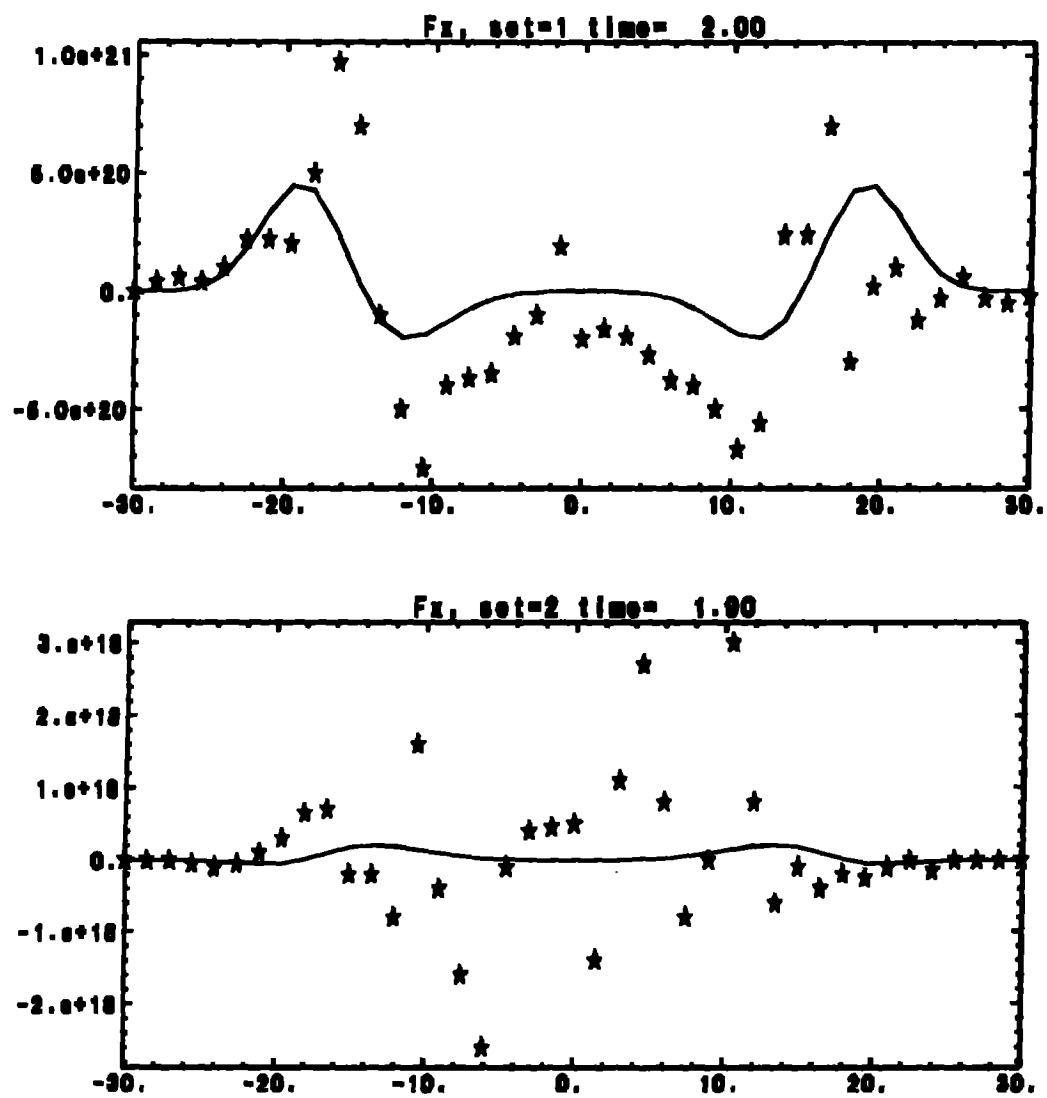


Figure 24:

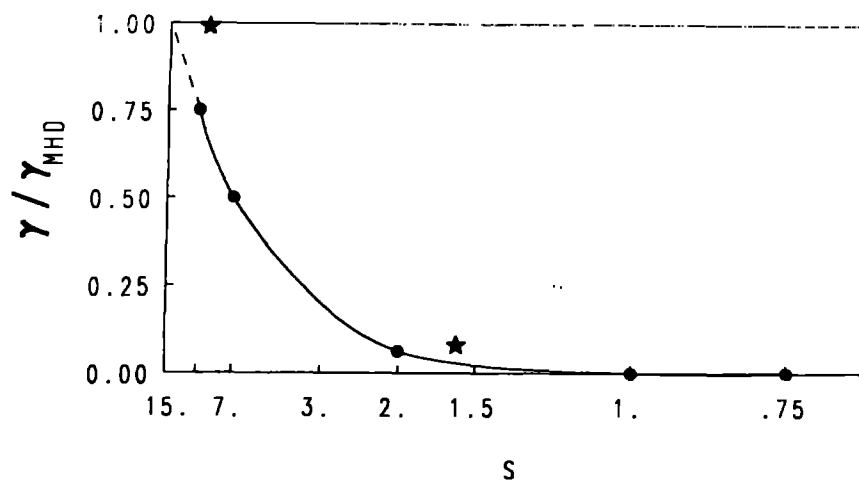


Figure 25: